A PIECEWISE LINEAR FINITE ELEMENT DISCRETIZATION OF THE DIFFUSION EQUATION

A Thesis

by

TERESA S. BAILEY

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2006

Major Subject: Nuclear Engineering

A PIECEWISE LINEAR FINITE ELEMENT DISCRETIZATION OF THE DIFFUSION EQUATION

A Thesis

by

TERESA S. BAILEY

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Marvin Adams
Committee Members,	Jean-Luc Guermond
	Jim Morel
	Paul Nelson
	Michael Zika
Head of Department,	William Burchill

August 2006

Major Subject: Nuclear Engineering

ABSTRACT

A Piecewise Linear Finite Element Discretization of the Diffusion Equation.

(August 2006)

Teresa S. Bailey, B.S., Oregon State University

Chair of Advisory Committee: Dr. Marvin Adams

In this thesis, we discuss the development, implementation and testing of a piecewise linear (PWL) continuous Galerkin finite element method applied to the three-dimensional diffusion equation. This discretization is particularly interesting because it discretizes the diffusion equation on an arbitrary polyhedral mesh. We implemented our method in the KULL software package being developed at Lawrence Livermore National Laboratory. This code previously utilized Palmer's method as its diffusion solver, which is a finite volume method that can produce an asymmetric coefficient matrix. We show that the PWL method produces a symmetric positive definite coefficient matrix that can be solved more efficiently, while retaining the accuracy and robustness of Palmer's method. Furthermore, we show that in most cases Palmer's method is actually a non-Galerkin PWL finite element method.

Because the PWL method is a Galerkin finite element method, it has a firm theoretical background to draw from. We have shown that the PWL method is a well-posed discrete problem with a second-order convergence rate. We have also performed a simple mode analysis on the PWL method and Palmer's method to compare the accuracy of each method for a certain class of problems.

Finally, we have run a series of numerical tests to uncover more properties of both the PWL method and Palmer's method. These numerical results indicate that the PWL method, partially due to its symmetric matrix, is able to solve large-scale diffusion problems very efficiently.

ACKNOWLEDGMENTS

I would first like to acknowledge the Computation Science Graduate Fellowship administered by the Krell Institute, which funded this research and my graduate education. I have always felt extremely fortunate to have received this fellowship. I would also like to thank Mike Zika and Brian Yang at Lawrence Livermore National Laboratory who gave me the opportunity to pursue this research and the assistance I needed to complete the project.

The other members of my committee also contributed greatly to the success of this research. Dr. Adams has been an excellent advisor and I am grateful for the opportunity to be his student. Dr. Guermond contributed many of the basic ideas in the analysis section of this thesis and I appreciate his patience in teaching me the mathematical theory behind this analysis. The other two members of my committee, Dr. Morel and Dr. Nelson, made many useful suggestions which were invaluable contributions to this work.

Finally, I would like to thank Dr. Palmer at Oregon State University who first introduced me to transport and diffusion theory, and recommended that I attend Texas A&M for graduate school. I doubt that I would be studying these subjects without the encouragement of Dr. Palmer.

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I
INTRODUCTION

Description of the problem

The diffusion equation is a mathematical model that can describe many physical processes including conductive heat transfer, atomic migration in materials, and photon and sub-atomic particle interaction with background materials. For this reason, great efforts have been made to solve the diffusion equation accurately and efficiently. Because practical problems are often too complicated for analytic solutions, many of these solutions are performed using numerical approximations. Despite years of research, improved numerical methods are still needed to provide more accurate and efficient diffusion solutions, especially on unstructured spatial meshes. Currently, new methods are being developed to spatially discretize the diffusion equation on a mesh whose cells are arbitrary polyhedra.

In this thesis, we introduce, analyze, implement, and numerically test a new method for discretizing the three-dimensional diffusion equation on arbitrary polyhedral grids. This new method is a Galerkin continuous finite element method that uses piecewise linear (PWL) basis functions. These functions were originally created by Stone and Adams and applied to the two-dimensional linear Boltzmann transport equation [1]. The fundamental idea behind the development of the PWL basis functions is to combine standard linear functions, or Lagrange functions, defined on tetrahedral subcells to form piecewise linear functions that span cells. Individual linear basis functions are easily written for each individual tetrahedral subcell. A certain weighted sum of these

---

This thesis follows the style of Journal of Computational Physics.

individual linear basis functions forms a PWL basis function over the entire cell. The weightings are carefully chosen so that the functions exactly interpolate linear functions on the polyhedral cells.

Because the PWL method we employ is a Galerkin finite element method, it will produce a symmetric positive definite coefficient matrix, which is much more computationally efficient to solve than an asymmetric matrix, and which requires less storage due to its symmetry. This symmetry is the main advantage of the method relative to asymmetric methods.

Other methods exist to solve the diffusion equation on arbitrary polyhedral grids. One approach to solving this problem is to divide the mesh into smaller hexahedral or tetrahedral cells and apply known finite element methods to the resulting meshes. Another approach is to use meshes made of polyhedral cells and apply a finite volume method on "dual-cell" volumes. This finite volume approach is known as Palmer's method [2,3]. Another approach is to develop a finite element approximation and use Wachspress rational basis functions, which have been shown to be an appropriate basis for meshes composed of "well-posed" polyhedral cells [4,5].

Each of the previous methods mentioned is a vertex-centered method. An advantage of a cell-centered method is that it can represent solution discontinuities at material interfaces more accurately than vertex-centered methods. A cell-centered finite volume method has been developed by Morel [6]. This method uses face unknowns to represent a cell average gradient. Another class of cell-centered methods, called support-operator methods, has been developed to solve the diffusion equation on arbitrary polyhedral grids [7].

In order to test the numerical properties of the PWL method, we have implemented it within the KULL software project at Lawrence Livermore National Laboratory. The

KULL project develops massively parallel, time dependent, three dimensional, multi-physics codes that attempt to simulate inertial confinement fusion systems. It spatially discretizes all physics equations on arbitrary polyhedral grids. In KULL, the diffusion solution can be used to model conductive heat transfer and radiation diffusion in inertial confinement fusion systems. The current method used by KULL to solve the diffusion equation is Palmer's method, which produces an asymmetric matrix in practical problems.

The radiation diffusion equation is an approximation of the radiation transport equation, which is a linear Boltzmann equation. The transport equation more accurately describes radiation movement through materials and space. Both of these equations model the particle or energy density in the system. The particle density becomes an energy density when it is multiplied by the particle energy. A density becomes an "intensity" or "angular flux" when it is multiplied by particle speed. The particle intensity solved for using the transport equation is dependent on the direction that photons are traveling, while the intensity solved for using the diffusion equation is not directionally dependent. Because the diffusion equation is not directionally dependent, it is computationally and numerically less challenging to solve than the transport equation. As a result, the diffusion equation is commonly used to approximate the movement of radiation through materials. In the next section we show a derivation of a diffusion equation from the transport equation for thermal radiation.

Diffusion as a limit of radiation transport

It is easy to show the relationship between the transport equation and the diffusion equation through an asymptotic analysis of the transport equation. For inertial confinement fusion problems, the time-dependent, mono-energetic transport equation is an energy balance equation for thermal radiation:

$$\frac{1}{c}\frac{\partial \psi}{dt}\left(\vec{r},\vec{\Omega},\nu,t\right)+\vec{\Omega}\bullet\vec{\nabla}\psi\left(\vec{r},\vec{\Omega},\nu,t\right)+\sigma\left(\vec{r},\nu,T\left(\vec{r}\right)\right)\psi\left(\vec{r},\vec{\Omega},\nu,t\right) = \sigma\left(\vec{r},\nu,T\left(\vec{r}\right)\right)B\left(\nu,T\left(\vec{r}\right)\right) \tag{1.1}$$

It is coupled to a differential equation describing temperature changes in the system through the Planckian term, *B*, which physically represents the emission of photons.

$$C_p\left(\vec{r},T\left(\vec{r}\right)\right)\frac{\partial T\left(\vec{r}\right)}{dt} = \int_0^\infty\int_{4\pi}\sigma\left(\vec{r},\nu,T\left(\vec{r}\right)\right)\left[\psi\left(\vec{r},\vec{\Omega},\nu,t\right)-B\left(\nu,T\left(\vec{r}\right)\right)\right]d\Omega d\nu+Q\left(\vec{r},t\right) \tag{1.2}$$

where the Planckian is defined as

$$B\left(\nu,T\left(\vec{r}\right)\right)=\frac{4\pi h\nu^3}{c^2}\frac{1}{e^{h\nu/kT(\vec{r})}-1}. \tag{1.3}$$

Another useful definition is the radiation energy density, which is defined as

$$E_R\left(\vec{r},t\right)=aT_R^4\left(\vec{r},t\right)=\frac{1}{c}\int_0^\infty d\nu\int_{4\pi}d\Omega\,\psi\left(\vec{r},\vec{\Omega},\nu,t\right), \tag{1.4}$$

where the radiation constant, a, is

$$a=\frac{8\pi^5k^4}{15h^3c^3}. \tag{1.5}$$

In the following asymptotic analysis, terms in both equations are scaled by a small number ε, as shown below. The scaling causes the absorption and re-emission of particles to be the dominant physical processes. The standard "equilibrium" diffusion equation is invariant under this scaling, which offers hope that the forthcoming analysis may produce this equation. See [8] for details. Here we are following the procedure outlined by Adams and Nowak [9].

$$\frac{\varepsilon}{c}\frac{\partial \psi}{dt}\left(\vec{r},\vec{\Omega},\nu,t\right)+\vec{\Omega}\bullet\vec{\nabla}\psi\left(\vec{r},\vec{\Omega},\nu,t\right)+\frac{\sigma\left(\vec{r},\nu,T\left(\vec{r}\right)\right)}{\varepsilon}\psi\left(\vec{r},\vec{\Omega},\nu,t\right) = \frac{\sigma\left(\vec{r},\nu,T\left(\vec{r}\right)\right)}{\varepsilon}B\left(\nu,T\left(\vec{r}\right)\right) \tag{1.6}$$

and

$$\varepsilon C_p\left(\vec{r},T\left(\vec{r}\right)\right)\frac{\partial T\left(\vec{r}\right)}{dt}$$
$$=\int_0^{\infty}\int_{4\pi}\frac{\sigma\left(\vec{r},\nu,T\left(\vec{r}\right)\right)}{\varepsilon}\left[\psi\left(\vec{r},\vec{\Omega},\nu,t\right)-B\left(\nu,T\left(\vec{r}\right)\right)\right]d\Omega d\nu+\varepsilon Q\left(\vec{r},t\right) \quad . \tag{1.7}$$

All variables dependent on position, ψ, T, σ, and B, are then expanded in terms of ε.

$$\begin{aligned}\psi\left(\vec{r},\vec{\Omega},\nu,t\right)&=\psi^{(0)}+\varepsilon\psi^{(1)}+\varepsilon^2\psi^{(2)}...\\ T\left(\vec{r},t\right)&=T^{(0)}+\varepsilon T^{(1)}+\varepsilon^2T^{(2)}...\\ \sigma\left(\vec{r},\nu,T\right)&=\sigma^{(0)}+\varepsilon\sigma^{(1)}+\varepsilon^2\sigma^{(2)}...\\ B\left(\nu,T\right)&=B^{(0)}+\varepsilon B^{(1)}+\varepsilon^2B^{(2)}...\end{aligned} \tag{1.8}$$

and substituted into Eqs. (1.6) and (1.7).

$$\begin{aligned}&\frac{\varepsilon}{c}\frac{\partial}{dt}\left(\psi^{(0)}+\varepsilon\psi^{(1)}+\varepsilon^2\psi^{(2)}...\right)+\vec{\Omega}\bullet\vec{\nabla}\left(\psi^{(0)}+\varepsilon\psi^{(1)}+\varepsilon^2\psi^{(2)}...\right)\\ &\quad+\frac{\left(\sigma^{(0)}+\varepsilon\sigma^{(1)}+\varepsilon^2\sigma^{(2)}...\right)}{\varepsilon}\left(\psi^{(0)}+\varepsilon\psi^{(1)}+\varepsilon^2\psi^{(2)}...\right)\\ &\quad=\frac{\left(\sigma^{(0)}+\varepsilon\sigma^{(1)}+\varepsilon^2\sigma^{(2)}...\right)}{\varepsilon}\left(B^{(0)}+\varepsilon B^{(1)}+\varepsilon^2B^{(2)}...\right)\end{aligned} \tag{1.9}$$

and

$$\varepsilon C_p\left(\vec{r},T\right)\frac{\partial}{dt}\left(T^{(0)}+\varepsilon T^{(1)}+\varepsilon^2T^{(2)}...\right)$$
$$=\int_0^{\infty}\int_{4\pi}\frac{\left(\sigma^{(0)}+\varepsilon\sigma^{(1)}+\varepsilon^2\sigma^{(2)}...\right)}{\varepsilon}\begin{bmatrix}\left(\psi^{(0)}+\varepsilon\psi^{(1)}+\varepsilon^2\psi^{(2)}...\right)\\ -\left(B^{(0)}+\varepsilon B^{(1)}+\varepsilon^2B^{(2)}...\right)\end{bmatrix}d\Omega d\nu+\varepsilon Q\left(\vec{r},t\right) \tag{1.10}$$

The O($1/\varepsilon$) terms are collected. From Eq. (1.9)

$$\frac{\sigma^{(0)}}{\varepsilon}\left(\psi^{(0)}\right)=\frac{\sigma^{(0)}}{\varepsilon}B^{(0)}\left(\nu,T^{(0)}\right), \tag{1.11}$$

resulting in

$$\psi^{(0)}=B^{(0)}\left(\nu,T^{(0)}\right). \tag{1.12}$$

Physically this result means that the leading-order intensity is the Planckian. Next, collect the O(1) terms in Eq. (1.9).

$$\vec{\Omega}\bullet\vec{\nabla}\psi^{(0)}+\left[\sigma\psi\right]^{(1)}=\left[\sigma B\right]^{(1)} \tag{1.13}$$

where

$$\begin{aligned}\left[\sigma\psi\right]^{(1)}&=\sigma^{(0)}\psi^{(1)}+\sigma^{(1)}\psi^{(0)}\\ \left[\sigma B\right]^{(1)}&=\sigma^{(0)}B^{(1)}+\sigma^{(1)}B^{(0)}\end{aligned}. \tag{1.14}$$

As a result,

$$\left[\sigma\psi\right]^{(1)}=-\vec{\Omega}\bullet\vec{\nabla}\psi^{(0)}+\left[\sigma B\right]^{(1)}. \tag{1.15}$$

We now define a quantity

$$\vec{J}=\int_{4\pi}d\Omega\vec{\Omega}\,\psi\left(\vec{\Omega}\right), \tag{1.16}$$

which is known as current. Find the O(ε) current by integrating Eq. (1.15) over all directions, Ω.

$$\left[\sigma\vec{J}\right]^{(1)}=-\int_{4\pi}d\Omega\vec{\Omega}\vec{\Omega}\bullet\vec{\nabla}\psi^{(0)}+\int_{4\pi}d\Omega\vec{\Omega}\left[\sigma B\right]^{(1)} \tag{1.17}$$

The result of these integrals is

$$\left[\sigma\vec{J}\right]^{(1)}=-\frac{1}{3}\vec{\nabla}\psi^{(0)}. \tag{1.18}$$

The integral involving the Planckian cancels because B is isotropic, and $\vec{\Omega}$ is an odd function. The next step is to expand the $\left[\sigma\vec{J}\right]^{(1)}$ term.

$$\left[\sigma\vec{J}\right]^{(1)}=\sigma^{(0)}\vec{J}^{(1)}+\sigma^{(1)}\vec{J}^{(0)}=-\frac{1}{3}\vec{\nabla}\psi^{(0)} \tag{1.19}$$

Note that

$$\vec{J}^{(0)}=\int_{4\pi}d\Omega\vec{\Omega}\,\psi^{(0)}=\int_{4\pi}d\Omega\vec{\Omega}B^{(0)}=0, \tag{1.20}$$

resulting in

$$J^{(1)}=-\frac{1}{3\sigma^{(0)}}\vec{\nabla}\psi^{(0)}=-\frac{1}{3\sigma^{(0)}}\vec{\nabla}B^{(0)}. \tag{1.21}$$

Collect the O(ε) terms in Eqs. (1.9) and (1.10). From Eq. (1.7)

$$\frac{1}{c}\frac{\partial}{dt}\left(\psi^{(0)}\right)+\vec{\Omega}\bullet\vec{\nabla}\left(\psi^{(1)}\right)+\left[\sigma\psi\right]^{(2)}=\left[\sigma B\right]^{(2)}. \tag{1.22}$$

When rearranged, Eq. (1.22) becomes

$$\left[\sigma\psi\right]^{(2)}=-\frac{1}{c}\frac{\partial}{dt}\left(\psi^{(0)}\right)-\vec{\Omega}\bullet\vec{\nabla}\left(\psi^{(1)}\right)+\left[\sigma B\right]^{(2)}. \tag{1.23}$$

From Eq. (1.10)

$$C_p\left(\vec{r},T\right)\frac{\partial}{dt}\left(T^{(0)}\right)=\int_0^\infty\int_{4\pi}\left[\sigma\psi\right]^{(2)}-\left[\sigma B\right]^{(2)}d\Omega d\nu+Q\left(\vec{r},t\right). \tag{1.24}$$

Substituting Eq. (1.23) into Eq. (1.24) results in

$$C_p\left(\vec{r},T\right)\frac{\partial}{dt}\left(T^{(0)}\right)=-\int_0^\infty\int_{4\pi}\frac{1}{c}\frac{\partial}{dt}\left(\psi^{(0)}\right)+\vec{\Omega}\bullet\vec{\nabla}\left(\psi^{(1)}\right)d\Omega d\nu+Q\left(\vec{r},t\right), \tag{1.25}$$

which can be simplified to

$$C_p\left(\vec{r},T\right)\frac{\partial}{dt}\left(T^{(0)}\right)+\frac{1}{c}\frac{\partial}{dt}\int_0^\infty\int_{4\pi}\left(\psi^{(0)}\right)d\Omega d\nu+\vec{\nabla}\bullet\int_0^\infty\int_{4\pi}\vec{\Omega}\,\psi^{(1)}d\Omega d\nu=Q\left(\vec{r},t\right). \tag{1.26}$$

Note that $\vec{J}^{(1)}=\int_{4\pi}\vec{\Omega}\,\psi^{(1)}d\Omega$, causing Eq. (1.26) to become

$$C_p\left(\vec{r},T\right)\frac{\partial}{dt}\left(T^{(0)}\right)+\frac{1}{c}\frac{\partial}{dt}\int_0^\infty\int_{4\pi}\left(\psi^{(0)}\right)d\Omega d\nu+\vec{\nabla}\bullet\int_0^\infty\vec{J}^{(1)}d\nu=Q\left(\vec{r},t\right). \tag{1.27}$$

The substitution of Eq. (1.21) into Eq. (1.27) for the current results in

$$\begin{aligned}&C_p\left(\vec{r},T\right)\frac{\partial}{dt}\left(T^{(0)}\right)+\frac{1}{c}\frac{\partial}{dt}\int_0^\infty\int_{4\pi}\left(\psi^{(0)}\right)d\Omega d\nu\\&-\vec{\nabla}\bullet\int_0^\infty\frac{1}{3\sigma\left(\nu\right)^{(0)}}\vec{\nabla}B\left(\nu,T\right)^{(0)}d\nu=Q\left(\vec{r},t\right)\end{aligned}. \tag{1.28}$$

The integral in the third term of Eq. (1.28) must be treated specially. This integral is

$$\begin{aligned}&\int_0^\infty\frac{1}{\sigma\left(\nu\right)^{(0)}}\vec{\nabla}B\left(\nu,T\right)^{(0)}d\nu\\&=\int_0^\infty\frac{1}{\sigma\left(\nu\right)^{(0)}}\left(\frac{\partial}{\partial x}B\left(\nu,T\right)^{(0)}\hat{i}+\frac{\partial}{\partial y}B\left(\nu,T\right)^{(0)}\hat{j}+\frac{\partial}{\partial z}B\left(\nu,T\right)^{(0)}\hat{k}\right)d\nu\end{aligned}. \tag{1.29}$$

Each derivative can be simplified using the chain rule. For example

$$\frac{\partial}{\partial x}B(\nu,T)^{(0)}\hat{i}=\left(\frac{\partial B}{\partial T^4}\right)\left(\frac{\partial T^4}{\partial x}\right)\hat{i}=\left(\frac{\partial B}{\partial T}\frac{\partial T}{\partial T^4}\right)\left(\frac{\partial T^4}{\partial x}\right)\hat{i}=\frac{1}{4T^3}\left(\frac{\partial B}{\partial T}\right)\left(\frac{\partial T^4}{\partial x}\right)\hat{i}\ . \quad (1.30)$$

If this chain rule is performed on each derivative in the integral the result is

$$\begin{aligned}&\int_0^\infty \frac{1}{\sigma(\nu)^{(0)}}\vec{\nabla}B(\nu,T)^{(0)}d\nu\\ &=\int_0^\infty \frac{1}{\sigma(\nu)^{(0)}}\frac{1}{4T^3}\left(\frac{\partial B}{\partial T}\right)\left(\left(\frac{\partial T^4}{\partial x}\right)\hat{i}+\left(\frac{\partial T^4}{\partial y}\right)\hat{j}+\left(\frac{\partial T^4}{\partial z}\right)\hat{k}\right)d\nu\end{aligned}\ , \quad (1.31)$$

which simplifies to

$$\int_0^\infty \frac{1}{\sigma(\nu)^{(0)}}\vec{\nabla}B(\nu,T)^{(0)}d\nu=\frac{1}{4T^3}\left[\vec{\nabla}\left(T^{(0)}\right)^4\right]\int_0^\infty \frac{1}{\sigma(\nu)^{(0)}}\left(\frac{\partial B}{\partial T}\right)d\nu\ . \quad (1.32)$$

We can multiply and divide Eq. (1.32) by $\int_0^\infty\left(\frac{\partial B}{\partial T}\right)d\nu$, resulting in

$$\int_0^\infty \frac{1}{\sigma(\nu)^{(0)}}\vec{\nabla}B(\nu,T)^{(0)}d\nu=\frac{1}{4T^3}\left[\vec{\nabla}\left(T^{(0)}\right)^4\right]\left[\frac{\int_0^\infty \frac{1}{\sigma(\nu)^{(0)}}\left(\frac{\partial B}{\partial T}\right)d\nu}{\int_0^\infty\left(\frac{\partial B}{\partial T}\right)d\nu}\right]\int_0^\infty\left(\frac{\partial B}{\partial T}\right)d\nu\ . \quad (1.33)$$

The term in the brackets in Eq. (1.33) is an "averaged" cross section value known as the Rosseland mean opacity [9].

$$\frac{1}{\sigma_R}\equiv\left[\frac{\int_0^\infty \frac{1}{\sigma(\nu)^{(0)}}\left(\frac{\partial B}{\partial T}\right)d\nu}{\int_0^\infty\left(\frac{\partial B}{\partial T}\right)d\nu}\right] \quad (1.34)$$

Using this definition, Eq. (1.29) becomes

$$\int_0^\infty \frac{1}{\sigma(\nu)^{(0)}}\vec{\nabla}B(\nu,T)^{(0)}d\nu=\frac{1}{4\sigma_R T^3}\left[\vec{\nabla}\left(T^{(0)}\right)^4\right]\frac{\partial}{\partial T}\int_0^\infty Bd\nu\ . \quad (1.35)$$

The integral of the Planckian over all frequencies is

$$\int_0^\infty Bd\nu=\int_0^\infty \frac{4\pi h\nu^3}{c^2}\frac{1}{e^{h\nu/kT}-1}d\nu=acT^4\ , \quad (1.36)$$

where the radiation constant, a, was previously defined in Eq. (1.5). When the two values from Eqs. (1.5) and (1.36) are inserted into Eq. (1.35), the result is

$$\int_0^\infty \frac{1}{\sigma(\nu)^{(0)}} \vec{\nabla} B(\nu,T)^{(0)} d\nu = \frac{1}{4\sigma_R T^3} \left[\vec{\nabla} \left(T^{(0)} \right)^4 \right] \frac{\partial}{\partial T} \left(acT^4 \right), \tag{1.37}$$

which can be simplified to

$$\int_0^\infty \frac{1}{\sigma(\nu)^{(0)}} \vec{\nabla} B(\nu,T)^{(0)} d\nu = \frac{ac}{\sigma_R} \left[\vec{\nabla} \left(T^{(0)} \right)^4 \right]. \tag{1.38}$$

From this result, Eq (1.28) becomes

$$C_p(\vec{r},T) \frac{\partial}{dt} \left(T^{(0)} \right) + \frac{1}{c} \frac{\partial}{dt} \int_0^\infty \int_{4\pi} \left(\psi^{(0)} \right) d\Omega d\nu - \vec{\nabla} \bullet \frac{ac}{3\sigma_R} \left[\vec{\nabla} \left(T^{(0)} \right)^4 \right] = Q(\vec{r},t). \tag{1.39}$$

Use the definition of the radiation density from Eq. (1.4) in Eq. (1.39).

$$C_p(\vec{r},T) \frac{\partial}{dt} \left(T^{(0)} \right) + a \frac{\partial}{dt} \left(T^{(0)} \right)^4 - \vec{\nabla} \bullet \frac{ac}{3\sigma_R} \left[\vec{\nabla} \left(T^{(0)} \right)^4 \right] = Q(\vec{r},t) \tag{1.40}$$

The first term of Eq. (1.29) is often neglected, because it is often slowly varying compared to the other terms resulting in a diffusion equation for T^4.

$$a \frac{\partial}{dt} \left(T^{(0)} \right)^4 - \vec{\nabla} \bullet \frac{ac}{3\sigma_R} \nabla \left(T^{(0)} \right)^4 = Q(\vec{r},t) \tag{1.41}$$

Overview of chapters

In this introductory chapter, we have briefly described physical problems that can be modeled using a diffusion equation. We introduced the PWL finite element method, and briefly described other methods that are able to solve the diffusion equation on an arbitrary polyhedral grid. We also gave a brief description of the software package KULL in which this new method was implemented, and the various diffusion systems that KULL models. The following chapters will include much more detail about the development and analysis of the PWL method.

Chapter II includes an in-depth derivation of the PWL finite element method. This derivation includes a general finite element discretization of the diffusion equation with a mathematical description of the PWL basis functions. The second chapter also includes a brief mathematical description of Palmer's method and descriptions of some other methods that can solve this diffusion problem, including each method's benefits and detriments.

Chapter III contains various mathematical proofs and analyses that help us characterize the advantages and disadvantages of both the PWL method and Palmer's method. It begins with a proof that the PWL finite element method produces a well-posed discrete problem. We also show a mathematical proof that the PWL method will have a second-order convergence rate. The next part of Chapter III is a mode analysis that allows us to compare the error of Palmer's discretization to the error of the PWL discretization for each Fourier mode in a specific set of problems. This analysis shows that a finite volume method (Palmer's method) will have a smaller error than a "lumped" finite element method (PWL) for a certain set of problems. This analysis also shows that the presence of high-aspect-ratio cells cause off-diagonal elements in the PWL coefficient matrix to become positive. Furthermore, it quantifies to some extent the loss of accuracy caused by lumping the PWL mass matrix.

In Chapter IV, we describe the implementation of the PWL method in KULL. This description includes a discussion about the coding required to build the coefficient matrix for a parallel calculation. It also contains a brief description of running test problems and building meshes for these test problems.

A description of various test problems and their results that compare the PWL method and Palmer's method can be found in Chapter V. These test problems include a series of problems with linear solutions; a series of problems to determine the convergence rates of each method; a problem with high-aspect-ratio cells and a highly localized source;

and a problem with radiation flow through a crooked duct to test the effectiveness of various linear solvers on both methods. We will conclude our discussion in Chapter VI by summarizing the properties of the PWL method that we have discovered in our analysis and testing and by suggesting future applications and extensions of this work.

CHAPTER II
METHOD DESCRIPTION

In this chapter, we present a formal description of the piecewise linear finite element method applied to the diffusion equation. This description includes the finite element derivation and a mathematical description of the PWL basis functions for arbitrary polyhedral cells. This chapter also includes brief descriptions of Palmer's method and the other methods that can be used as discretizations of the diffusion equation on arbitrary polyhedral cells.

Finite element method (FEM) applied to diffusion on arbitrary polyhedral grids

In Chapter I, we derived a time dependent diffusion equation, which we will use in test our method.

$$a\frac{\partial}{dt}\left(T^{(0)}\left(\vec{r}\right)\right)^4-\vec{\nabla}\bullet\frac{ac}{3\sigma_R}\nabla\left(T^{(0)}\left(\vec{r}\right)\right)^4=Q\left(\vec{r},t\right) \tag{2.1}$$

For simplicity write this as

$$a\frac{\partial}{dt}E\left(\vec{r}\right)-\vec{\nabla}\bullet D\vec{\nabla}E\left(\vec{r}\right)=Q\left(\vec{r},t\right), \tag{2.2}$$

where

$$E\left(\vec{r}\right)=T^4\left(\vec{r}\right) \tag{2.3}$$

and

$$D=\frac{ac}{3\sigma_R}. \tag{2.4}$$

We begin discretizing Eq. (2.2) by applying a simple fully-implicit Euler approximation to the time derivative, which results in

$$a\frac{\left(E^{n+1}\left(\vec{r}\right)-E^{n}\left(\vec{r}\right)\right)}{\Delta t}-\vec{\nabla}\bullet D\vec{\nabla}E^{n+1}\left(\vec{r}\right)=Q^{n}\left(\vec{r}\right), \tag{2.5}$$

where the n superscripts represent a time step. In Eq. (2.5), solve for E^{n+1} to get a diffusion equation.

$$-\vec{\nabla}\bullet D\vec{\nabla}E^{n+1}\left(\vec{r}\right)+\sigma E^{n+1}\left(\vec{r}\right)=S^{n}\left(\vec{r}\right), \tag{2.6}$$

where

$$\sigma=\frac{a}{\Delta t} \tag{2.7}$$

and

$$S^{n}\left(\vec{r}\right)=Q^{n}\left(\vec{r}\right)+\frac{a}{\Delta t}E^{n}\left(\vec{r}\right). \tag{2.8}$$

Eq. (2.6) is the final form of the steady state diffusion equation to which we will apply all spatial discretizations. We also define a net current, which will be a useful quantity in this derivation. The diffusion net current is

$$\vec{F}=-D\vec{\nabla}E. \tag{2.9}$$

To apply a continuous FEM to the diffusion equation, we first multiply the equation by a weight function and integrate it over the problem domain. If the i-th weight function is nonzero only on cells that touch the i-th vertex, then we have

$$\int_{z\,at\,i} d^{3}r\, w_{i}\left(r\right)\left[-\vec{\nabla}\bullet D\vec{\nabla}E\left(\vec{r}\right)+\sigma E\left(\vec{r}\right)-S\left(\vec{r}\right)\right]=0, \tag{2.10}$$

where z denotes cells (which we sometimes call “zones”). The divergence theorem produces:

$$\int_{\partial z at\, i} d^2 r\, w_i(\vec{r})\vec{n}\bullet\vec{F} + \int_{z at\, i} d^3 r \left[D\vec{\nabla}E(\vec{r})\bullet\vec{\nabla}w_i(\vec{r}) + w_i(\vec{r})\left(\sigma E(\vec{r}) - S(\vec{r})\right)\right] = 0 . \quad (2.11)$$

If the i-th weight function goes to zero on cell surfaces that do not touch vertex i the first term is zero if vertex i is not on the boundary of the domain. We presently ignore it. An approximation can be made for E in terms of known basis functions:

$$E(\vec{r}) = \sum_{all\, j} E_j b_j(\vec{r}), \quad (2.12)$$

which results in

$$\sum_j E_j \left\{ \int_{z at\, i} d^3 r \left[D\vec{\nabla}b_j(\vec{r})\bullet\vec{\nabla}w_i(\vec{r}) + \sigma b_j(\vec{r}) w_i(\vec{r})\right]\right\} = \int_{z at\, i} d^3 r\, w_i(\vec{r}) S(\vec{r}) . \quad (2.13)$$

Eq. (2.13) is N equations for N unknowns, where N is the number of weight functions and the number of basis functions. Given our assumption of one weight function per vertex, N is the number of vertices in the mesh, and the number of rows in the corresponding coefficient matrix.

To incorporate boundary conditions into a continuous finite element discretization of the diffusion equation, recall, from Eq. (2.11), the term

$$\int_{\partial z at\, i} d^2 r\, w_i(\vec{r})\vec{n}\bullet D\vec{\nabla}E . \quad (2.14)$$

This term was neglected because it was non-zero only for the vertices on the boundary. Three types of boundary conditions exist in our problems of interest: Neumann boundary conditions, Dirichlet boundary conditions, and mixed boundary conditions. The Neumann boundary conditions, often referred to as natural boundary conditions, are the simplest boundary conditions to implement. These boundary conditions define a value of the gradient of the unknown function on the boundary, which allows us to move the boundary term to the right hand side of the equation and treat it like a source. If the

gradient value is zero, then the boundary condition is reflecting and is incorporated into the method naturally. For this reason, reflecting boundary conditions are the default boundary conditions in the implementation of this discretization.

Dirichlet boundary conditions define a value of the unknown function at the boundary vertices. In order to implement these boundary conditions, we ignore the weighted-integral equation for the boundary vertices and simply set up the coefficient matrix such that the vertex value of the unknown is set to be equal to its Dirichlet boundary value. In this case, the row of the matrix corresponding to the boundary vertex has the number one as the diagonal value and zero for other column values. Its source vector value is the value set by the boundary condition. Adding Dirichlet boundary conditions to the discretization will make the coefficient matrix asymmetric unless every other row of the matrix that has a non-zero value corresponding to a Dirichlet boundary vertex is changed to compensate for the asymmetry caused by the boundary condition.

Mixed, or Robin, boundary conditions are often used in diffusion problems to represent known inflow. These boundary conditions have the form of

$$C = aE + b\vec{n} \bullet D\vec{\nabla}E \,, \tag{2.15}$$

where E is the unknown on the boundary, a and b are any constant coefficients, and C is the value of the boundary condition. To implement these boundary conditions, we solve for the gradient of the unknown in Eq. (2.15), which results in

$$\vec{n} \bullet D\vec{\nabla}E = \frac{C}{b} - \frac{a}{b}E \tag{2.16}$$

and then substitute this value into Eq. (2.14). The result of this manipulation is

$$\int_{\partial z at i} d^2 r\, w_i(r) \vec{n} \bullet D\vec{\nabla}E = -E_i \int_{\partial z at i} d^2 r\, w_i(r) \frac{a}{b} + \int_{\partial z at i} d^2 r\, w_i(r) \frac{C}{b}. \tag{2.17}$$

Eq. (2.17) is then substituted into Eq. (2.11), resulting in

$$-E_i \int_{\partial z\, at\, i} d^2 r\, w_i(r) \frac{a}{b} + \int_{\partial z\, at\, i} d^2 r\, w_i(r) \frac{C}{b}$$
$$+ \int_{z\, at\, i} d^3 r \left[D\vec{\nabla} E(\vec{r}) \bullet \vec{\nabla} w_i(\vec{r}) + w_i(\vec{r}) \left(\sigma E(\vec{r}) - S(\vec{r}) \right) \right] = 0 \qquad (2.18)$$

To incorporate these integrals into the matrix, the integral in the first term of Eq. (2.18) is added to the diagonal element of the *i-th* row, and the value of the second integral is added to the *i-th* value in the source vector.

In this work we develop a Galerkin FEM that employs the piecewise linear (PWL) weight and basis functions developed recently by Stone and Adams [4]. In the development of these methods for arbitrary polyhedral cells, we divide each polyhedron into subcells called sides, corners, and wedges. A side is a tetrahedron made from two adjacent vertices, the zone center, and a face center. A corner, which will be used primarily in the development of Palmer's method, is defined as the union of all half-sides that touch a vertex in one zone. A wedge is defined to be a half side. See Figures 2.1, 2.2, and 2.3 for depictions of side subcells, corner subcells, and wedges in a hexahedral cell.

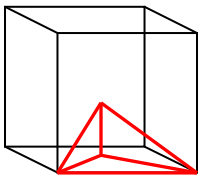


Figure 2.1: Side in a hexahedral cell

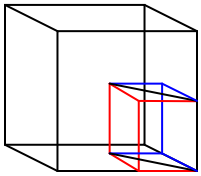


Figure 2.2: Corner in a hexahedral cell

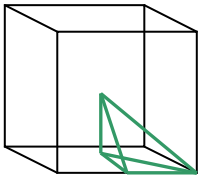


Figure 2.3: Wedge in a hexahedral cell

In these three-dimensional cells, it is possible to have a face whose vertices are non-coplanar. In order to create sides on this face, we define a face center point and facet the curved surface about this face center. This concept is illustrated in Figure 2.4. This figure shows a four-sided face of a three-dimensional arbitrary polyhedral cell.

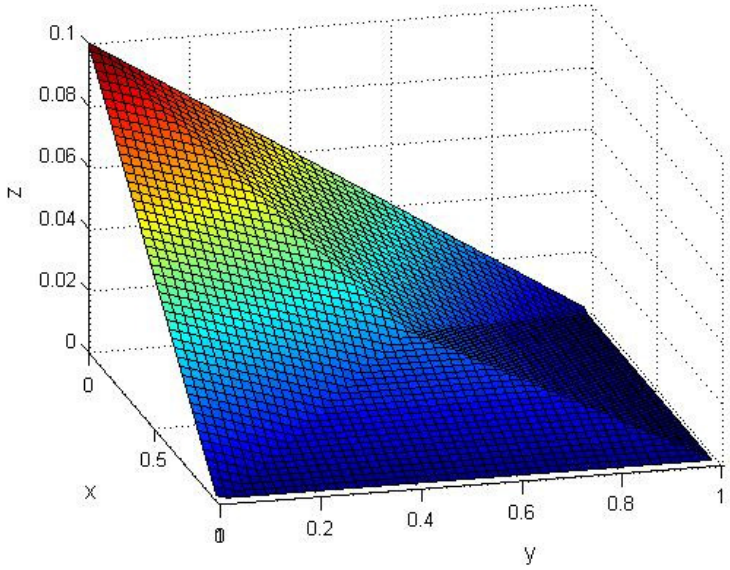


Figure 2.4: Sides on a face with non-coplanar vertices

Because the weight and basis functions are known and are simple polynomials on each "side" subcell, all integrations in Eq. (2.13) can be performed analytically over each side of a given cell:

$$\sum_{all\ j} E_j \sum_{sides} \left(\int_{V_{side}} d^3 r D_{side} \left(\vec{\nabla} w_i \right)_{side} \bullet \left(\vec{\nabla} b_j \right)_{side} + \sigma_{side} \int_{V_{side}} d^3 r\ w_i b_j \right) = \sum_{all\ j} S_j \sum_{sides} \int_{V_{side}} d^3 r\ w_i b_j \ . \quad (2.19)$$

Because the method is Galerkin, which means that $w_i = b_i$, it is easy to see that it will produce a symmetric matrix for interior cells. If *i* and *j* are interchanged in Eq. (2.19), the equation will not change, meaning that the *i,j* element of the matrix has the same value as the *j,i* value of the matrix. The matrix rows that represent boundary vertices can also be forced to be symmetric. In general, the matrix produced by a Galerkin FEM is symmetric positive definite (SPD), which is easy to show.

Proof that Galerkin FEMs produce SPD matrices

A matrix is positive definite if $\forall x \neq 0\ x^T A x > 0$. This inequality can be written as

$$\sum_{i=1}^{N} \sum_{j=1}^{N} x_i A_{i,j} x_j > 0 \quad (2.20)$$

Inserting the value of $A_{i,j}$ and noting that $w_i = b_i$, the inequality in Eq. (2.20) becomes

$$\sum_{i=1}^{N} \sum_{j=1}^{N} x_i \left(\int d^3 r D \vec{\nabla} b_i(\vec{r}) \bullet \vec{\nabla} b_j(\vec{r}) + \sigma \int d^3 r\ b_i(\vec{r}) b_j(\vec{r}) \right) x_j > 0 \ . \quad (2.21)$$

The integrals and summations can be rearranged, resulting in

$$\int d^3 r D \sum_{i=1}^{N} \sum_{j=1}^{N} x_i \vec{\nabla} b_i(\vec{r}) \bullet \vec{\nabla} b_j(\vec{r})\, x_j + \int d^3 r \sigma \sum_{i=1}^{N} \sum_{j=1}^{N} x_i b_i(\vec{r}) b_j(\vec{r})\, x_j > 0 \ . \quad (2.22)$$

The arguments in the summations in Eq. (2.22) can be then separated.

$$\int d^3 r D \sum_{i=1}^{N} \left[x_i \vec{\nabla} b_i(\vec{r}) \bullet \sum_{j=1}^{N} \vec{\nabla} b_j(\vec{r}) x_j \right] + \int d^3 r \sigma \sum_{i=1}^{N} \left[x_i b_i(\vec{r}) \sum_{j=1}^{N} b_j(\vec{r}) x_j \right] > 0 \qquad (2.23)$$

Set the sums over *j* to be equal to

$$\begin{aligned} \vec{g}(\vec{r}) &= \sum_{j=1}^{N} \left[x_j \vec{\nabla} b_j(\vec{r}) \right] \\ h(\vec{r}) &= \sum_{j=1}^{N} \left[x_j b_j(\vec{r}) \right] \end{aligned} . \qquad (2.24)$$

Note that in Eq. (2.24)

$$\vec{g}(\vec{r}) = \vec{\nabla} h(\vec{r}) \qquad (2.25)$$

When the definitions in Eq. (2.24) are substituted into Eq. (2.23), the result is

$$\int d^3 r D \vec{g}(\vec{r}) \bullet \sum_{i=1}^{N} \left[x_i \vec{\nabla} b_i(\vec{r}) \right] + \int d^3 r \sigma h(\vec{r}) \sum_{i=1}^{N} \left[x_i b_i(\vec{r}) \right] > 0 . \qquad (2.26)$$

Note that the remaining summations are also equal to the definitions in Eq. (2.24) because the summation over *i* is over the same range as the summation over *j*. When these definitions are substituted into Eq. (2.26), the result is

$$\int d^3 r D \vec{g}(\vec{r}) \bullet \vec{g}(\vec{r}) + \int d^3 r \sigma h^2(\vec{r}) > 0 \qquad (2.27)$$

In most mathematical cases, the inequality in Eq. (2.27) holds. The second term in Eq. (2.27) is always greater than zero if σ is non-zero, but can be zero if σ is zero. The value of *h* in this term can be constant if a linear combination of the basis functions sums to a constant. This is equivalent to a constant solution. If *h* is constant, then *g*, which is the gradient of *h*, will be zero, causing the first term in Eq. (2.27) to be equal to zero. As a result, it is mathematically possible for the inequality in Eq. (2.27) to not be true when σ is zero and the solution (linear combination of the basis functions) is constant, which makes *g* in Eq. (2.24) go to zero. However, this case does not make physical sense because this case corresponds to a steady state problem with reflecting boundary

conditions where there are no loss mechanisms. If there are no loss mechanisms in a steady state problem, there can be no source mechanisms either, which means that we have generated a problem with no particles, a hence a problem with a zero solution. We can now say that the inequality in Eq. (2.27) is true if we neglect the unphysical case where σ is zero and the solution is constant. This result means that any Galerkin FEM is SPD for all physical steady state problems.

Definition of piecewise linear (PWL) basis functions

To define our specific Galerkin finite element method, we need to define our basis functions. The PWL basis function centered at vertex *j* can be written in three dimensions as:

$$b_j(\vec{r}) = t_j(\vec{r}) + \sum_{faces\ at\ j} \beta_{f,j} t_f(\vec{r}) + \alpha_{z,j} t_z(\vec{r}), \tag{2.28}$$

where the *t* functions are standard linear functions defined tetrahedron by tetrahedron. (Recall that each "side" subcell is a tetrahedron formed by the zone center, a face center, and two adjacent vertices on that face.) For example, t_j equals 1 at the *j*-th vertex and decreases linearly to zero on all other vertices of each side that touches point *j*. t_z is unity at the cell midpoint and zero at each face midpoint and each cell vertex. t_f is unity at the face midpoint and zero at the cell midpoint and at each of the face's vertices. The α_z and β_f are weights that give the cell and face midpoints as weighted averages of their vertices:

$$\vec{r}_z \equiv \text{ cell midpoint} = \sum_{j@z} \alpha_{z,j} \vec{r}_j ; \tag{2.29}$$

$$\vec{r}_f \equiv \text{ face midpoint} = \sum_{j@f} \beta_{f,j} \vec{r}_j . \tag{2.30}$$

The weights are normalized to unity. To show this property, we take the sum of the basis functions over a face of a cell. This sum must be equal to one everywhere on the face of the cell

$$1 = \sum_{j=1}^{N} b_j(\vec{r}) = \sum_{j=1}^{N} t_j(\vec{r}) + \sum_{j=1}^{N} \beta_{f,j} t_f(\vec{r}) + \alpha_{z,j} t_z(\vec{r}), \tag{2.31}$$

where N is the number of vertices on the face. We now evaluate Eq. (2.31) at the face center, where all t_j and t_z will be zero by definition. This evaluation causes Eq. (2.31) to simplify to

$$1 = t_f(\vec{r}) \sum_{j=1}^{N} \beta_{f,j} . \tag{2.32}$$

We also know that, at the face center point, $t_f(\vec{r})$, must be equal to one, by definition. As a result,

$$1 = \sum_{j=1}^{N} \beta_{f,j} \tag{2.33}$$

on a face of the cell. To prove that the cell center weights, α_j, must be normalized to one, we take the sum of the basis functions over the entire cell, which must equal one everywhere in the cell.

$$1 = \sum_{j=1}^{N} b_j(\vec{r}) = \sum_{j=1}^{N} t_j(\vec{r}) + \sum_{j=1}^{N} \sum_{f @ j} \beta_{f,j} t_f(\vec{r}) + \sum_{j=1}^{N} \alpha_{z,j} t_z(\vec{r}), \tag{2.34}$$

where N is the number of vertices in the cell. We now evaluate Eq. (2.34) at the cell center, where all t_j and t_f will be zero by definition. This evaluation causes Eq. (2.34)to simplify to

$$1 = t_z(\vec{r}) \sum_{j=1}^{N} \alpha_{z,j} \tag{2.35}$$

Again, by definition , $t_z(\vec{r})$ is one at the cell center, resulting in

$$1 = \sum_{j=1}^{N} \alpha_{z,j} \,, \tag{2.36}$$

for a given cell.

Each basis function is linear on each side, which makes integration over sides straightforward. (Note that the gradient of a basis function is constant on a side.) Figure 2.5 shows a plot of a PWL basis function for a two-dimensional rectangular cell.

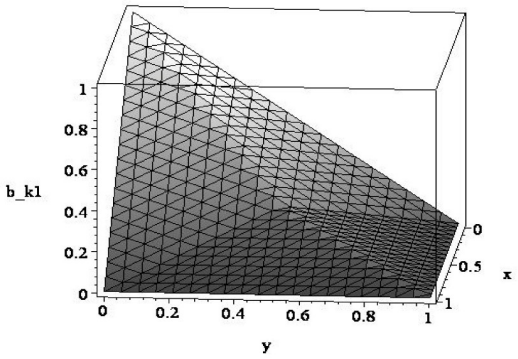


Figure 2.5: Two-dimensional piecewise linear basis function

Figure courtesy of Hiromi Stone [1]

The definitions in Eqs. (2.28)-(2.30)guarantee that any linear function of (*x,y,z*) can be exactly represented as an expansion in these basis functions. (This subtlety is what makes the method work. Effectively, the functions generate interpolated values at cell and face midpoints, in such a way that the interpolated values are perfect for linear functions. The importance of this property becomes clear in a convergence rate analysis in Chapter III.)

Our Galerkin FEM uses PWL functions for the *w* and *b* in the general equation above. In our implementation in the KULL project code, however, we use a certain kind of lumping of the "mass matrix." That is, we make the replacement:

$$\int_{z@i} d^3r \left[\sigma b_j(r) w_i(r)\right] \xrightarrow{lump} \delta_{ij} \int_{c@i} d^3r \left[\sigma\right], \tag{2.37}$$

where *c* refers to a "corner" subcell. This defines our method except for boundary conditions, which have previously been discussed.

Derivation of Palmer's method

Palmer's method is very similar to the PWL method. Palmer's method was originally derived as a vertex-centered finite-volume method [2,3]. The control volume over which the diffusion equation is integrated is the union of all of the wedges surrounding a given vertex. This volume is often called a dual cell, and is shown on a two dimensional orthogonal mesh in Figure 2.6.

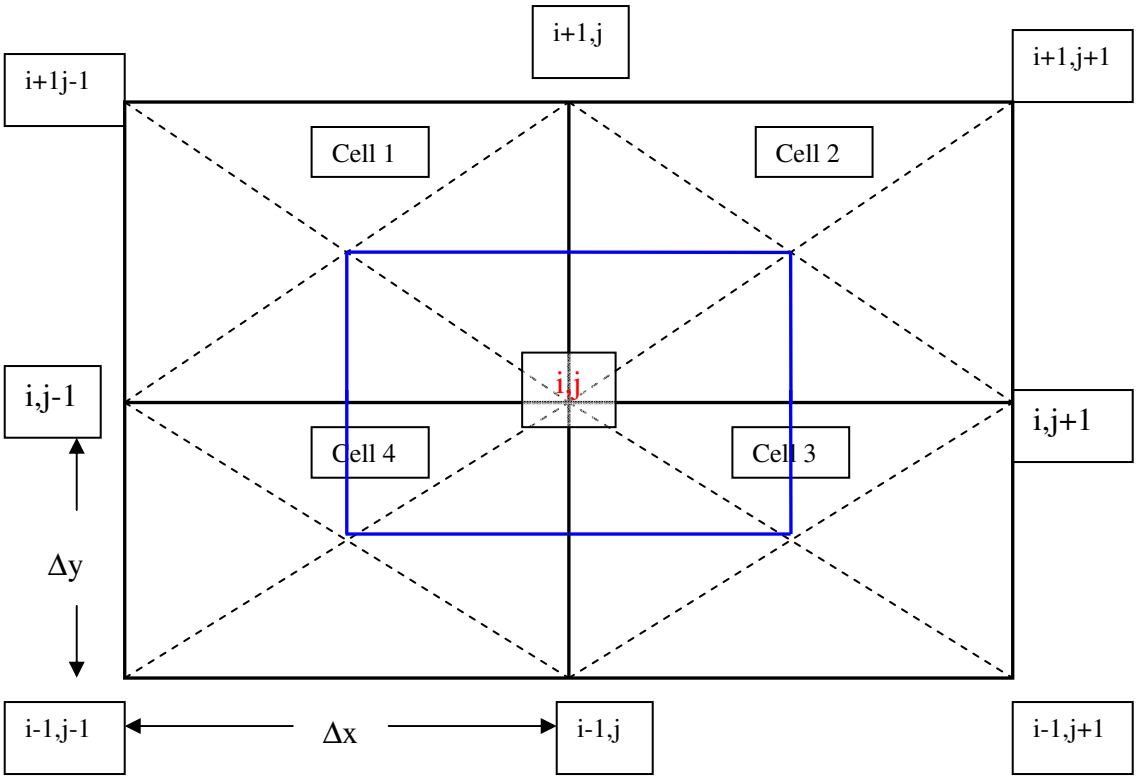


Figure 2.6: Dual-cell finite volume in two dimensions

The finite volume integration of the diffusion equation is shown in Eq. (2.38) .

$$\int_{V_{\text{Dual Cell}}} \left[\vec{\nabla}\bullet\vec{F}\left(\vec{r}\right)+\sigma E\left(\vec{r}\right)-S\left(\vec{r}\right)\right]d^3r=0 \tag{2.38}$$

The first term in this integral can be simplified using the Gauss Divergence theorem resulting in

$$\int_{\partial V_{\text{Dual Cell}}} \vec{n}\bullet\vec{F}\left(\vec{r}\right)ds+\int_{V_{\text{Dual Cell}}} \left[\sigma E\left(\vec{r}\right)-S\left(\vec{r}\right)\right]d^3r=0\,. \tag{2.39}$$

In order to evaluate the volume integrals, the flux, *E*, and the source *S* is assumed to be the values at the vertex. This assumption results in terms that are equivalent to those obtained by mass-matrix lumping in the PWL finite element method. We can also rewrite the surface integral as a sum over the surfaces of the dual cell.

$$\sum_{\substack{Surfaces\\ \in Dual\ Cell}} \vec{A}\bullet\vec{F}\left(\vec{r}\right)+E_p\sum_{\substack{wedges\\ \in Dual\ Cell}} \sigma_w V_w=S_p\sum_{\substack{wedges\\ \in Dual\ Cell}} V_w \tag{2.40}$$

where the subscript *p* indicates the vertex corresponding to the dual cell, and the subscript *w* indicates the wedges connected to the vertex *p*. Note also that $\vec{F}\left(\vec{r}\right)=-D\vec{\nabla}E\left(\vec{r}\right)$, which when substituted into Eq. (2.40) results in

$$-\sum_{\substack{Surfaces\\ \in Dual\ Cell}} \vec{A}\bullet D_w\left[\vec{\nabla}E\left(\vec{r}\right)\right]_w+E_p\sum_{\substack{wedges\\ \in Dual\ Cell}} \sigma_w V_w=S_p\sum_{\substack{wedges\\ \in Dual\ Cell}} V_w\,. \tag{2.41}$$

The remaining task is to represent the gradient of *E* in terms of the vertex values. Note that the gradients are needed only on the planes that divide sides into equal half-sides (wedges)—these are the planes that form the surfaces of the dual cell. Each side is a tetrahedron (3D) or triangle (2D). If *E* were known at each vertex of each side, this would define a unique constant gradient in each side, and the task would be complete.

However, one vertex of the side is a cell midpoint (and in 3D one is a face midpoint). We do not have values of E at these points; our E unknowns are only at cell vertices.

The gist of Palmer's method is to *interpolate* to get E values at cell midpoints (and at face midpoints in 3D) from the cell-vertex values. In general, such interpolations can be written as:

$$E_z \equiv \text{ cell midpoint} = \sum_{p@z} \alpha_{z,p} E_p \tag{2.42}$$

$$E_f \equiv \text{ face midpoint} = \sum_{p@f} \beta_{f,p} E_p \tag{2.43}$$

Palmer recognized the need for these interpolations to be exact for linear functions and defined the interpolation weights, cell midpoints, and face midpoints to ensure this property. Although this method was developed prior to the "invention" of PWL basis functions, it is equivalent to using them. (Palmer defined cell and face midpoint coordinates as simple averages of vertex coordinates and defined the α and β parameters accordingly as 1/N. The developers of PWL recognized that other choices are possible as long as Eqs. (2.29) and (2.30) are respected.)

Given the interpolated z and f values, it is a simple matter to construct gradients on each side and thus on each dual-cell surface. These gradients are easily expressed in terms of the cell-vertex values of E by use of Eqs. (2.42) and (2.43). This completes the method.

This method for a non-orthogonal logically rectangular mesh in 2D has a nine-point stencil. That is, each vertex is coupled to all vertices in all the cells that touch it. For a 2D rectangular mesh, Palmer's discretization produces the simple five-point stencil. In the simple case of uniform mesh spacing and constant material properties this becomes:

$$2D\left(\frac{\Delta y}{\Delta x}+\frac{\Delta x}{\Delta y}\right)\phi_{i,j}-D\left(\frac{\Delta y}{\Delta x}\right)\left(\phi_{i,j-1}+\phi_{i,j+1}\right)-D\left(\frac{\Delta x}{\Delta y}\right)\left(\phi_{i+1,j}+\phi_{i-1,j}\right) \\ +\sigma_a\phi_{i,j}\Delta x\Delta y=S_{i,j}\Delta x\Delta y \qquad (2.44)$$

Palmer's method is equivalent to a Petrov-Galerkin PWL FEM

It is easy to show that Palmer's method is actually a Petrov-Galerkin continuous finite element method. This equivalent FEM has PWL basis functions and weight functions that are constant in the dual cell and zero elsewhere. However, this equivalence is only true if the diffusion coefficient is constant in each side subcell. To show this we return to the general weighted-integral equation derived previously:

$$\int_{\partial z\,at\,i} ds\, w_i(\vec{r})\vec{n}\bullet\vec{F}+\int_{z\,at\,i} d^2r\left[D\vec{\nabla}E(\vec{r})\bullet\vec{\nabla}w_i(\vec{r})\right] \\ +\int_{z\,at\,i} d^2r\left[w_i(\vec{r})\left(\sigma E(\vec{r})-S(\vec{r})\right)\right]=0 \qquad (2.45)$$

Notice that the second term in Eq. (2.45) is zero because the weight functions are constant, and the gradient of a constant is zero, reducing this equation to

$$\int_{\partial z\,at\,i} ds\, w_i(\vec{r})\vec{n}\bullet\vec{F}+\int_{z\,at\,i} d^2r\left[w_i(\vec{r})\left(\sigma E(\vec{r})-S(\vec{r})\right)\right]=0. \qquad (2.46)$$

Also, notice that the integral over the surface of the cells can be reduced to a summation.

$$\sum_{\substack{Surfaces\\ \in Dual\ cell}} \vec{A}\bullet\vec{F}(r)+\int_{z\,at\,i} d^2r\left[w_i(r)\left(\sigma E(r)-S(r)\right)\right]=0. \qquad (2.47)$$

If mass matrix lumping is used in this Petrov-Galerkin finite element method, the result of the remaining integrals is

$$\sum_{\substack{Surfaces\\ \in Dual\ Cell}} \vec{A}\bullet\vec{F}(\vec{r})+E_p\sum_{\substack{wedges\\ \in Dual\ Cell}}\sigma_w V_w=S_p\sum_{\substack{wedges\\ \in Dual\ Cell}} V_w \qquad (2.48)$$

Eq. (2.48) is equivalent to Eq. (2.40), which means that Palmer's method is equivalent to a Petrov-Galerkin PWL method if the approximation for the current is the same. Palmer's method forces the gradient of E to be constant in each side of each cell, which means E in each side is forced to be linear. The PWL basis functions are linear by definition, so the same approximation is made in each side of each cell for both methods. Furthermore, the center point about which the PWL basis functions are defined and the E_z value in Palmer's method are interpolated using the same value of α. A similar statement holds for the E_f (face-midpoint) value in 3D. For these reasons, Palmer's method is really a Petrov-Galerkin finite element method with PWL basis functions and with weight functions that are constant in the dual cell volumes, as long as the diffusion coefficient is constant in each side.

One last interesting connection between Palmer's finite volume method and the PWL finite element method is that they both reduce to a linear continuous Galerkin finite element method on triangular (2D) and tetrahedral (3D) grids if mass-matrix lumping is used in the FEMs. Stone and Adams have shown this for the PWL basis functions [1], and Palmer has shown this for Palmer's method [2]. The PWL result follows from the fact that the PWL functions reduce to standard linear functions given polygons that are triangles or polyhedra that are tetrahedral. Palmer's result follows from this and from the fact that each linear weight function integrates to the same value, which is the volume of the associated "corner" subcell.

Other methods with potential to solve the diffusion equation on arbitrary polyhedral grids

We have previously mentioned other strategies for solving the diffusion equation on arbitrary polyhedral grids. One such method is to divide the arbitrary polyhedral cells into tetrahedra and use "classical" finite element basis functions on these tetrahedra. The benefit of this approach is that it has a sound and well-research mathematical

background. From a practical point of view, we know this method will work. Furthermore, we can develop high-order basis functions that go beyond linear solution approximations. However, by dividing the mesh only into tetrahedra, we may add a considerable number of unnecessary unknowns, which can make the matrices more expensive to solve (relative to the PWL method). If we divide the mesh into tetrahedral cells without adding unknowns, then we will add asymmetry to grids and problems that began with certain symmetries.

Another approach is to use a different basis, known as Wachspress rational functions, in the Galerkin finite element method [4,5]. These basis functions are ratios of polynomials, and are developed such that they collapse to the trilinear continuous basis functions on orthogonal grids. Wachspress basis functions, unlike PWL basis functions, have continuous derivatives, meaning they can potentially represent solution curvature better than PWL basis functions. Also, Wachspress basis functions have the benefit of being able to work on elements with curved surfaces. However, they do not perform well as the angle between edges in a polygonal cell approaches 180 degrees. Furthermore, Wachspress basis functions cannot be integrated analytically, and their numerical integrations can make constructing the matrix coefficients computationally expensive.

Two other methods exist to solve the diffusion equation on arbitrary polyhedral meshes: Morel's method [6] and the Support Operator method [7]. These methods are cell-centered, which means their unknowns are located at the face centers and cell centers. In general, cell-centered methods can better model source and material discontinuities as compared to vertex-centered methods. Morel's method develops coupling between a cell-center unknown and the face center unknowns of that cell through a finite-difference approximation of a gradient of flux on the face. The gradients on the faces are defined such that a linear solution is preserved, and the method is locally conservative. However, the number of unknowns in Morel's method is significantly larger than in vertex-

centered methods. Also, for arbitrary polyhedral grids, Morel's method produces an asymmetric coefficient matrix.

The Support Operator method is similar to Morel's method. It is locally conservative, and has unknowns located at face and cell centers. However, it produces a symmetric positive definite coefficient matrix, and makes finite-volume-type approximations for the gradients in the cell. It has been numerically shown that the Support Operators method is second-order accurate, but this method requires as many unknowns as Morel's method, which is significantly more unknowns than vertex-centered methods require. In addition, it appears to be relatively expensive to compute the matrix coefficients for this method because the method requires additional algebra to write the cell-center unknown in terms of the face-center unknowns. Furthermore, this method requires a matrix inversion, where the size of the matrix is equal to the number of the faces in the cell, to solve for the current in each cell.

It is not clear that any single method will be superior for all diffusion problems on arbitrary grids. Further testing and analysis is required to determine which methods are "better" for different classes of problems, different physical applications, and perhaps for different computer architectures.

In this chapter we have developed a general finite element formulation applied to the diffusion equation. We then described the Piecewise Linear basis functions that we used in our finite element discretization. Palmer's method was then described. We showed that Palmer's method is equivalent to a Petrov-Galerkin PWL finite element method, whose weight functions are constant on dual cells. Finally, we briefly described other discretizations which have been developed to solve the diffusion equation on arbitrary polyhedral grids.

CHAPTER III
ANALYSIS OF PWL AND RELATED METHODS

In this chapter we perform some simple mathematical analyses of various methods applied to the diffusion equation to uncover some of their properties. We begin by proving that the PWL finite element method results in a well-posed numerical problem. We then show that the PWL solution converges to the analytic solution, in a certain norm, with O(h^2) error as the mesh is refined. Finally, we perform a simple mode analysis in two dimensions on lumped and unlumped versions of the PWL method, on Palmer's method, and on lumped and unlumped versions of a bilinear continuous finite element method (BLC). From this we draw inferences about the expected accuracy of these methods for certain problems.

Proof of well-posedness

We can prove many mathematical properties of the PWL method. First we can show that the application of this method to the diffusion equation results in a well-posed numerical problem using the Lax-Milgram Lemma [10]. We can write the diffusion equation as

$$\begin{aligned} -\Delta u + \lambda u &= f \\ u\,|_{\partial\Omega} &= 0 \end{aligned}, \tag{3.1}$$

where λ is a coefficient that represents the square of an inverse diffusion length, f is the source divided by the diffusion coefficient, and Ω is the domain of the problem. This problem is a homogeneous, one dimensional diffusion problem with Dirichlet vacuum boundary conditions, often referred to as essential boundary conditions.

We can write this diffusion equation in a weak formulation. First multiply the diffusion equation by a weight (or test) function, v, and integrate over the domain of the problem.

$$\int_{\Omega} d\Omega v\left[-\Delta u+\lambda v u\right]=\int_{\Omega} d\Omega v f \tag{3.2}$$

In Eq. (3.2), *u* represents the basis (trial) function. The first term in Eq. (3.2) can be simplified using the Gauss-Divergence theorem.

$$-\int_{\partial\Omega} v\vec{n}\bullet\vec{\nabla}u+\int_{\Omega} d\Omega\vec{\nabla}v\bullet\vec{\nabla}u+\lambda\int_{\Omega} d\Omega v u=\int_{\Omega} d\Omega v f \tag{3.3}$$

The weight function, *v*, is defined such that it is zero on the boundaries of the domain, causing the first term in Eq. (3.3) to disappear. Our weak form of this diffusion problem becomes:

$$\begin{aligned}&\int_{\Omega} d\Omega\vec{\nabla}v\bullet\vec{\nabla}u+\lambda\int_{\Omega} d\Omega v u=\int_{\Omega} d\Omega v f\\&u\in H_0^1(\Omega)\ \text{and}\ \forall\ v\in H_0^1(\Omega)\end{aligned} \tag{3.4}$$

where, $H_0^1(\Omega)$ is a Hilbert space and a norm can be defined such that

$$\|w\|_{H_0^1(\Omega)}=\left(\|\nabla w\|_{L^2}^2+\|w\|_{L^2}^2\right)^{1/2}. \tag{3.5}$$

For simplicity let

$$\begin{aligned}&a(u,v)\equiv\int_{\Omega} d\Omega\vec{\nabla}v\bullet\vec{\nabla}u+\lambda\int_{\Omega} d\Omega v u\\&S(v)\equiv\int_{\Omega} d\Omega v f\end{aligned}\ . \tag{3.6}$$

The Lax-Milgram Lemma states that a problem is well-posed (a solution exists and is unique) if

1. The weight and basis (or test and trial) functions are in a Hilbert space, H
2. *a(u,v)* is a bilinear form on H
3. *a(u,v)* is bounded

4. *a(u,u)* is coercive
5. *S(v)* is linear and bounded

The first two elements of the lemma have already been shown. By definition the weight and basis functions exist in a Hilbert space because it is known that $H_0^1(\Omega)$ is a Hilbert space.

In order to prove the third part of the Lax-Milgram Lemma, we start by taking the absolute value of *a(u,v)*, and writing the integrals in terms of norms.

$$\left|a(u,v)\right| \le \int_\Omega d\Omega \left|\vec{\nabla}u \bullet \vec{\nabla}v\right| + \left|\lambda\right| \int_\Omega d\Omega \left|uv\right| \tag{3.7}$$

$$\left|a(u,v)\right| \le \left\|\nabla u\right\|_{L^2} \left\|\nabla v\right\|_{L^2} + \left|\lambda\right| \left\|u\right\|_{L^2} \left\|v\right\|_{L^2} \tag{3.8}$$

Applying the Cauchy-Schwarz inequality results in

$$\begin{aligned} \left|a(u,v)\right| &\le \left(\left\|\nabla u\right\|_{L^2}^2 + \left|\lambda\right| \left\|u\right\|_{L^2}^2\right)^{1/2} \left(\left\|\nabla v\right\|_{L^2}^2 + \left|\lambda\right| \left\|v\right\|_{L^2}^2\right)^{1/2} \\ &\le (1+\lambda)\left(\left\|\nabla u\right\|_{L^2}^2 + \left\|u\right\|_{L^2}^2\right)^{1/2} \left(\left\|\nabla v\right\|_{L^2}^2 + \left\|v\right\|_{L^2}^2\right)^{1/2} \end{aligned} . \tag{3.9}$$

The final result of this algebra is:

$$\left|a(u,v)\right| \le (1+\lambda)\left(\left\|\nabla u\right\|_{L^2}^2 + \left\|u\right\|_{L^2}^2\right)^{1/2} \left(\left\|\nabla v\right\|_{L^2}^2 + \left\|v\right\|_{L^2}^2\right)^{1/2} \tag{3.10}$$

Substituting the definition for the H_0^1 norm given in Eq. (3.5) into Eq. (3.10) results in

$$\left|a(u,v)\right| \le (1+\lambda)\left\|u\right\|_{H^1(\Omega)} \left\|v\right\|_{H^1(\Omega)} . \tag{3.11}$$

We can now say that *a(u,v)* is bounded because

$$\frac{\left|a(u,v)\right|}{\left\|u\right\|_{H^1(\Omega)} \left\|v\right\|_{H^1(\Omega)}} \le (1+\lambda) < \infty \tag{3.12}$$

whenever *u* and *v* are non-zero.

The next step in applying the Lax-Milgram Lemma is to prove that the bilinear form *a* is coercive.

$$a(u,u) = \int_{\Omega} d\Omega \vec{\nabla} u \bullet \vec{\nabla} u + \lambda \int_{\Omega} d\Omega (u)^2 \tag{3.13}$$

Using the definition of the L^2 norm,

$$a(u,u) = \|\nabla u\|_{L^2}^2 + \lambda \|u\|_{L^2}^2 \geq \alpha \left\{ \|\nabla u\|_{L^2}^2 + \|u\|_{L^2}^2 \right\},\ \alpha = \min(1,\lambda), \lambda > 0 \tag{3.14}$$

Apply the H_0^1 norm defined by Eq.(3.5), resulting in

$$a(u,u) \geq \alpha \|u\|_{H^1(\Omega)}^2 \tag{3.15}$$

As long as this value of α does not become too large, *a* will be coercive.

The final step in proving the lemma is to prove that *S(v)* is bounded.

$$\begin{aligned} & S(v) = \int_{\Omega} d\Omega v f \\ & |S(v)| = \int_{\Omega} d\Omega |v||f| \\ & |S(v)| \leq \|f\|_{L^2} \|v\|_{L^2} \\ & \sup_{v \neq 0} \frac{|S(v)|}{\|v\|_{L^2(\Omega)}} \leq \|f\|_{L^2(\Omega)} < \infty \end{aligned} \tag{3.16}$$

The steps shown in Eq. (3.16) are similar to proving boundedness of *a(u,v)*. *S(v)* will be bounded only if the original source function is bounded in the L^2 norm. For our problems of interest this will always be true.

Because every condition of the Lax-Milgram lemma has been met, we have mathematically shown that the PWL FEM discretization and any other linear Galerkin continuous FEM will produce a well-posed numerical problem.

Derivation of general convergence properties

Using the results from the Lax-Milgram proof, we can also determine some properties about the rate at which the PWL solution converges to the analytic solution in the L_2 norm as the mesh is refined. To derive this convergence rate, we begin with the analytic solution, u, and a discrete solution, u_h, where

$$a\left(u, v_h\right) = \int_{\Omega} f v_h d\Omega \quad \forall v_h \in X_h\left(\Omega\right) \tag{3.17}$$

and

$$a\left(u_h, v_h\right) = \int_{\Omega} f v_h d\Omega \quad \forall v_h \in X_h\left(\Omega\right). \tag{3.18}$$

X_h is a collection of all weight and basis functions. From the coercivity condition in Eq. (3.15), we know that

$$C_1 \left\| u_h - w_h \right\|^2_{H^1(\Omega)} \le a\left(u_h - w_h, u_h - w_h\right) \forall w_h \in X_h \subset H^1_0\left(\Omega\right) \tag{3.19}$$

The left hand side of this inequality can be expanded by adding and subtracting u, the exact solution to the problem.

$$C_1 \left\| u_h - w_h \right\|^2_{H^1(\Omega)} \le a\left(u_h - w_h, u_h - w_h\right) = a\left(u_h - u + u - w_h, u_h - w_h\right), \tag{3.20}$$

which reduces to

$$C_1 \left\| u_h - w_h \right\|^2_{H^1(\Omega)} \le a\left(u_h - u, u_h - w_h\right) + a\left(u - w_h, u_h - w_h\right). \tag{3.21}$$

The subtraction of Eq. (3.17) from Eq. (3.18) results in

$$a\left(u_h - u, v_h\right) = 0 \quad \forall v_h \in X_h, \tag{3.22}$$

which when substituted in Eq. (3.21), results in

$$C_1 \left\| u_h - w_h \right\|^2_{H^1(\Omega)} \le a\left(u - w_h, u_h - w_h\right) \tag{3.23}$$

because $u_h - w_h \in X_h$. Now, we can use the boundedness property of *a* to rewrite Eq. (3.23) as

$$C_1 \left\| u_h - w_h \right\|^2_{H^1(\Omega)} \le a\left(u - w_h, u_h - w_h\right) \le C_2 \left\| u - w_h \right\|_{H^1(\Omega)} \left\| u_h - w_h \right\|_{H^1(\Omega)}, \quad (3.24)$$

which simplifies to

$$\left\| u_h - w_h \right\|_{H^1(\Omega)} \le C_3 \left\| u - w_h \right\|_{H^1(\Omega)}. \quad (3.25)$$

We can use the triangle inequality to simplify Eq. (3.25)

$$\left\| u - u_h \right\|_{H^1(\Omega)} \le \left\| u - w_h \right\|_{H^1(\Omega)} + \left\| w_h - u_h \right\|_{H^1(\Omega)} \quad \forall w_h \in X_h \quad (3.26)$$

Substituting Eq.(3.26) into Eq. (3.25) results in

$$\left\| u - u_h \right\|_{H^1(\Omega)} \le \left\{1 + C_3\right\} \left\| u - w_h \right\|_{H^1(\Omega)}. \quad (3.27)$$

This result can be generalized to

$$\left\| u - u_h \right\|_{H^1(\Omega)} \le \left\{1 + C_3\right\} \left\| u - I_h u \right\|_{H^1(\Omega)} \quad (3.28)$$

where $I_h u$ is an interpolation of *u* at all vertices in the problem. Thus, the error in an FEM solution (left side Eq. (3.28)) is closely related to the error made when its basis functions interpolate the solution (right side of Eq. (3.28)). From the Bramble-Hilbert Lemma, we know that if this interpolation function is the standard linear basis functions on a tetrahedral mesh, the method has a second-order convergence rate in the L_2 norm.

Proof of second-order convergence rate for PWL

First, we consider two-dimensional problems. As described in Chapter II, for a given polygonal shaped cell, if the cell is divided into triangular side subcell volumes, the usual triangular basis function interpolation can be written as

$$I_h u = \sum_{j \in h} t_j^L u\left(\vec{r}_j\right) + t_z^L u\left(\vec{r}_z\right) \quad (3.29)$$

where t_j^L is the standard linear basis function defined on the triangular side at point j and t_z^L is the sum of all standard linear basis functions whose support point is at the center of the polygon. For the same polygon, the PWL interpolation can be written as

$$\tilde{I}_h u = \sum_{j\in h} t_j^L u\left(\vec{r}_j\right) + t_z^L \sum_{j\in h} \alpha_j u\left(\vec{r}_j\right). \tag{3.30}$$

We can prove that the convergence rate of the PWL discretization will be second order if we can prove that the difference between the standard interpolation and the PWL interpolation is also second order. To do this, we first find the difference between Eq. (3.29) and Eq. (3.30).

$$\tilde{I}_h u - I_h u = t_z^L \left(\sum_{j\in h} \alpha_j u\left(\vec{r}_j\right) - u\left(\vec{r}_z\right) \right) \tag{3.31}$$

The important term in Eq. (3.31) is

$$\left(\sum_{j\in h} \alpha_j u\left(\vec{r}_j\right) - u\left(\vec{r}_z\right) \right). \tag{3.32}$$

If this term is a second-order difference then the PWL method will also have a second-order convergence rate.

We now perform a Taylor series expansion on $u\left(\vec{r}_j\right)$ about point $\vec{r}_z$

$$u\left(\vec{r}_j\right) = u\left(\vec{r}_z\right) + \left(\vec{r}_j - \vec{r}_z\right)\bullet\vec{\nabla} u\left(\vec{r}_z\right) + O\left(h^2\right). \tag{3.33}$$

If Eq. (3.33) is substituted into expression (3.32), the result is

$$\left(\sum_{j\in h} \alpha_j \left[u\left(\vec{r}_z\right) + \left(\vec{r}_j - \vec{r}_z\right)\bullet\vec{\nabla} u\left(\vec{r}_z\right) + O\left(h^2\right) \right] - u\left(\vec{r}_z\right) \right), \tag{3.34}$$

which simplifies to

$$\left(\sum_{j\in h} \alpha_j - 1 \right) u\left(\vec{r}_z\right) + \left(\sum_{j\in h} \alpha_j \left(\vec{r}_j - \vec{r}_z\right) \right)\bullet\vec{\nabla} u\left(\vec{r}_z\right) + O\left(h^2\right). \tag{3.35}$$

If

$$\sum_{j\in h}\alpha_j = 1 \tag{3.36}$$

and

$$\sum_{j\in h}\alpha_j\left(\vec{r}_j - \vec{r}_z\right) = 0 \Rightarrow \vec{r}_z = \sum_{j\in h}\alpha_j\vec{r}_j \tag{3.37}$$

then we see that the entire expression is $O(h^2)$. Thus, $\tilde{I}_h u - I_h u = O\left(h^2\right)$, which means the difference between PWL and a standard linear FEM on triangular subcells is $O(h^2)$. Since the standard linear FEM has $O(h^2)$ error, it follows that PWL does also.

We turn next to three dimensions. For a given polyhedral shaped cell (3D), if the cell is divided into sides, this interpolation can be written as:

$$I_h u = \sum_{j\in h} t_j^L u\left(\vec{r}_j\right) + \sum_{faces\in h} t_f^L u\left(\vec{r}_f\right) + t_z^L u\left(\vec{r}_z\right) \tag{3.38}$$

where t_j^L is the standard linear basis function defined on the tetrahedron at point j; t_f^L is the sum of all standard linear basis functions whose support point is at the center of a polyhedral face, f; and t_z^L is all standard linear basis functions whose support point is at the center of the polyhedron. These basis functions are described in more detail in Chapter II. For the same polyhedron, the PWL interpolation can be written as

$$\tilde{I}_h u = \sum_{j\in h} t_j^L u\left(\vec{r}_j\right) + \sum_{faces\in h} t_f^L \sum_{i\in face} \beta_{f,i} u\left(\vec{r}_i\right) + t_z^L \sum_{j\in h} \alpha_j u\left(\vec{r}_j\right). \tag{3.39}$$

Following the same procedure as the 2D case, we subtract Eq. (3.38) from Eq. (3.39).

$$\tilde{I}_h u - I_h u = \sum_{faces\in h} t_f^L \left(\sum_{i\in face} \beta_{f,i} u\left(\vec{r}_i\right) - u\left(\vec{r}_f\right) \right) + t_z^L \left(\sum_{j\in h} \alpha_j u\left(\vec{r}_j\right) - u\left(\vec{r}_z\right) \right) \tag{3.40}$$

If the difference in Eq. (3.40) is second order then the PWL method will be second-order convergent. This will be true if

$$\sum_{i\in face} \beta_{f,i} u\left(\vec{r}_i\right) - u\left(\vec{r}_f\right) = O\left(h^2\right) \tag{3.41}$$

and

$$\sum_{j\in h} \alpha_j u\left(\vec{r}_j\right) - u\left(\vec{r}_z\right) = O\left(h^2\right). \tag{3.42}$$

We perform Taylor series expansions on $u\left(\vec{r}_i\right)$ about point $\vec{r}_f$ and $u\left(\vec{r}_j\right)$ about point $\vec{r}_z$ resulting in

$$u\left(\vec{r}_i\right) = u\left(\vec{r}_f\right) + \left(\vec{r}_i - \vec{r}_f\right)\bullet\nabla u\left(\vec{r}_f\right) + O\left(h^2\right) \tag{3.43}$$

and

$$u\left(\vec{r}_j\right) = u\left(\vec{r}_z\right) + \left(\vec{r}_j - \vec{r}_z\right)\bullet\nabla u\left(\vec{r}_z\right) + O\left(h^2\right) \tag{3.44}$$

Eqs. (3.43) and (3.44) are then substituted into Eqs. (3.41) and (3.42) respectively, resulting in

$$\sum_{i\in face} \beta_{f,i}\left[u\left(\vec{r}_f\right) + \left(\vec{r}_i - \vec{r}_f\right)\bullet\nabla u\left(\vec{r}_f\right) + O\left(h^2\right)\right] - u\left(\vec{r}_f\right) \tag{3.45}$$

and

$$\sum_{j\in h} \alpha_j\left[u\left(\vec{r}_z\right) + \left(\vec{r}_j - \vec{r}_z\right)\bullet\nabla u\left(\vec{r}_z\right) + O\left(h^2\right)\right] - u\left(\vec{r}_z\right). \tag{3.46}$$

Expressions (3.45) and (3.46) simplify to

$$\left(\sum_{i\in face} \beta_{f,i} - 1\right) u\left(\vec{r}_f\right) + \left(\sum_{i\in face} \beta_{f,i}\left(\vec{r}_i - \vec{r}_f\right)\right)\bullet\nabla u\left(\vec{r}_f\right) + O\left(h^2\right) \tag{3.47}$$

and

$$\left(\sum_{i\in face} a_j - 1\right) u\left(\vec{r}_z\right) + \left(\sum_{i\in face} a_j\left(\vec{r}_j - \vec{r}_z\right)\right)\bullet\nabla u\left(\vec{r}_z\right) + O\left(h^2\right). \tag{3.48}$$

The summation terms will vanish, and hence the PWL method in 3D will be second order, if for every face we have:

$$\sum_{i\in f} \beta_{f,i} = 1 \tag{3.49}$$

and

$$\sum_{i\in f} \beta_{f,i} \left(\vec{r}_i - \vec{r}_f \right) = 0 ; \tag{3.50}$$

and for every cell we have

$$\sum_{j\in h} \alpha_j = 1 \tag{3.51}$$

and

$$\sum_{j\in h} \alpha_j \left(\vec{r}_j - \vec{r}_z \right) = 0 . \tag{3.52}$$

These are in fact the equations that the PWL method uses to determine its cell and face midpoints and its basis functions. Thus, we expect the PWL solution to converge like $O(h^2)$, in the norm described above, as the mesh is refined in 3D problems.

A simple mode analysis of PWL, Palmer's method, and a bilinear continuous (BLC) FEM

Here we analyze a simple test problem: infinite medium, 2D, rectangular cells, constant material properties, uniform mesh spacings. A mathematical analysis can be performed for this problem to determine what ratio of Δx to Δy results in sign changes in the coupled matrix elements, and to determine how the various methods approach the analytic solution in the limit as $\Delta x \to 0$ and $\Delta y \to 0$.

The analytic solution of the diffusion equation can be determined for a source that can be written in terms of Fourier modes. The diffusion equation is:

$$-D\nabla^2\phi+\sigma_a\phi=S\,, \tag{3.53}$$

which can be rewritten as

$$-D\left[\frac{d^2\phi(\vec{r})}{dx^2}+\frac{d^2\phi(\vec{r})}{dy^2}+\frac{d^2\phi(\vec{r})}{dz^2}\right]+\sigma_a\phi(\vec{r})=S(\vec{r})\,. \tag{3.54}$$

The source can be expanded as

$$S(\vec{r})=\int_{-\infty}^{\infty}dw_z\int_{-\infty}^{\infty}dw_y\int_{-\infty}^{\infty}dw_x S(\vec{w})\exp(i\vec{w}\bullet\vec{r}) \tag{3.55}$$

The solution to the diffusion equation can also be expanded:

$$\phi(\vec{r})=\int_{-\infty}^{\infty}dw_z\int_{-\infty}^{\infty}dw_y\int_{-\infty}^{\infty}dw_x\phi(\vec{w})\exp(i\vec{w}\bullet\vec{r})\,. \tag{3.56}$$

We need to find an expression of $\phi(w)$ to have a meaningful representation of the solution. In order to do this, substitute Eqs. (3.55) and (3.56) into Eq. (3.54), multiply by $\exp(-i\vec{w}'\bullet\vec{r})$, and integrate over all space.

$$\begin{aligned}
&-Di^2\int_{-\infty}^{\infty}dw_z\int_{-\infty}^{\infty}dw_y\int_{-\infty}^{\infty}dw_x\int_{-\infty}^{\infty}dz\int_{-\infty}^{\infty}dy\int_{-\infty}^{\infty}dx\phi(w)\exp\left(i\left[\vec{w}-\vec{w}'\right]\bullet\vec{r}\right)\left[w_x^2+w_y^2+w_z^2\right]\\
&\quad+\sigma_a\int_{-\infty}^{\infty}dw_z\int_{-\infty}^{\infty}dw_y\int_{-\infty}^{\infty}dw_x\int_{-\infty}^{\infty}dz\int_{-\infty}^{\infty}dy\int_{-\infty}^{\infty}dx\left[\phi(w)\exp\left(i\left[\vec{w}-\vec{w}'\right]\bullet\vec{r}\right)\right]\\
&\quad=\int_{-\infty}^{\infty}dw_z\int_{-\infty}^{\infty}dw_y\int_{-\infty}^{\infty}dw_x\int_{-\infty}^{\infty}dz\int_{-\infty}^{\infty}dy\int_{-\infty}^{\infty}dx\left[S(w)\exp\left(i\left[\vec{w}-\vec{w}'\right]\bullet\vec{r}\right)\right]
\end{aligned} \tag{3.57}$$

Eq. (3.57) can be expressed as

$$
\begin{aligned}
&-Di^2\int_{-\infty}^{\infty}dw_z\int_{-\infty}^{\infty}dw_y\int_{-\infty}^{\infty}dw_x\int_{-\infty}^{\infty}dz\,e^{i(w_z-w'_z)z}\int_{-\infty}^{\infty}dy\,e^{i(w_y-w'_y)y}\int_{-\infty}^{\infty}dx\left\{\phi(w)e^{i(w_x-w'_x)x}\left[w_x^2+w_y^2+w_z^2\right]\right\}\\
&\quad+\sigma_a\int_{-\infty}^{\infty}dw_z\int_{-\infty}^{\infty}dw_y\int_{-\infty}^{\infty}dw_x\int_{-\infty}^{\infty}dz\,e^{i(w_z-w'_z)z}\int_{-\infty}^{\infty}dy\,e^{i(w_y-w'_y)y}\int_{-\infty}^{\infty}dx\left[\phi(w)e^{i(w_x-w'_x)x}\right] \qquad (3.58)\\
&\quad=\int_{-\infty}^{\infty}dw_z\int_{-\infty}^{\infty}dw_y\int_{-\infty}^{\infty}dw_x\int_{-\infty}^{\infty}dz\,e^{i(w_z-w'_z)z}\int_{-\infty}^{\infty}dy\,e^{i(w_y-w'_y)y}\int_{-\infty}^{\infty}dx\left[S(w)e^{i(w_x-w'_x)x}\right]
\end{aligned}
$$

It is well known that

$$\frac{1}{2\pi}\int_{-\infty}^{\infty}dx\,e^{i(w-w')x}=\delta(w-w') \qquad (3.59)$$

Using the relationship found in Eq. (3.59), Eq. (3.58) can be rewritten as

$$
\begin{aligned}
&-Di^2\int_{-\infty}^{\infty}dw_z\int_{-\infty}^{\infty}dw_y\int_{-\infty}^{\infty}dw_x\left\{\phi(w)\delta(w_z-w'_z)\delta(w_y-w'_y)\delta(w_x-w'_x)\left[w_x^2+w_y^2+w_z^2\right]\right\}\\
&\quad+\sigma_a\int_{-\infty}^{\infty}dw_z\int_{-\infty}^{\infty}dw_y\int_{-\infty}^{\infty}dw_x\left[\phi(w)\delta(w_z-w'_z)\delta(w_y-w'_y)\delta(w_x-w'_x)\right] \qquad (3.60)\\
&\quad=\int_{-\infty}^{\infty}dw_z\int_{-\infty}^{\infty}dw_y\int_{-\infty}^{\infty}dw_x\left[S(w)\delta(w_z-w'_z)\delta(w_y-w'_y)\delta(w_x-w'_x)\right]
\end{aligned}
$$

Using properties of the delta function, the simplification of Eq. (3.60) is

$$-Di^2\left[w_x^2+w_y^2+w_z^2\right]\phi_w\exp(i\vec{w}\bullet\vec{r})+\sigma_a\phi_w\exp(i\vec{w}\bullet\vec{r})=S_w\exp(i\vec{w}\bullet\vec{r}) \qquad (3.61)$$

This shows that the Fourier modes which we used to express the source and the solution are linearly independent. For this reason we can solve for each mode independently:

$$\phi_w=\frac{S_w}{D\|w\|^2+\sigma_a}. \qquad (3.62)$$

The result found in Eq. (3.62) is the analytic solution for each solution mode given its corresponding source mode.

The next step in this analysis is to determine how various methods behave when applied to this type of problem. For each numerical method we will perform the analysis only

for a two-dimensional problem on a rectangular grid. Figure 3.1 shows a reference grid whose numbering will be used in the analysis for all methods.

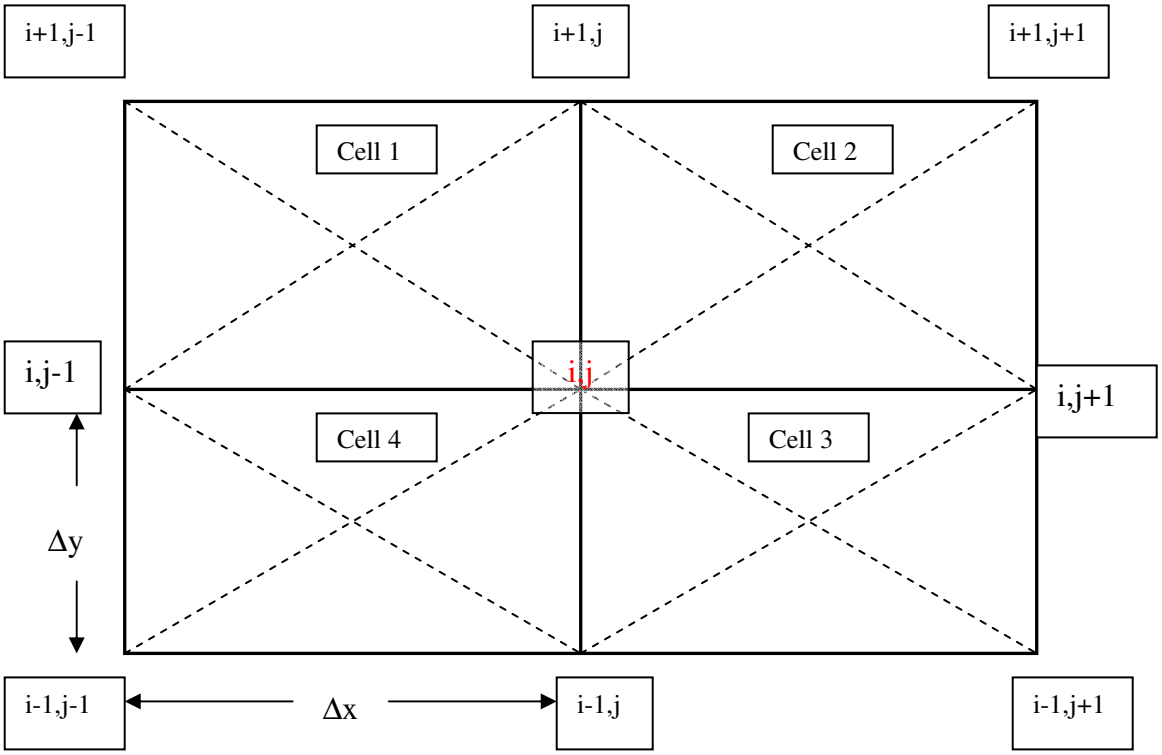


Figure 3.1: Two-dimensional rectangular reference grid

The first method we will examine is the mass-lumped continuous piecewise linear finite element method. The result of this spatial discretization for one row of the coefficient matrix produces a nine-point coupling and is found in Eq. (3.63).

$$
\begin{aligned}
&\frac{3}{2}D\left[\frac{(\Delta x)^2+(\Delta y)^2}{(\Delta x)(\Delta y)}\right]\phi_{i,j} \\
&+\frac{1}{8}D\left[\frac{(\Delta y)^2-3(\Delta x)^2}{(\Delta x)(\Delta y)}\right]\left(\phi_{i+1,j}+\phi_{i+1,j}+\phi_{i-1,j}+\phi_{i-1,j}\right) \\
&+\frac{1}{8}D\left[-\frac{(\Delta x)^2+(\Delta y)^2}{(\Delta x)(\Delta y)}\right]\left(\phi_{i+1,i-1}+\phi_{i+1,i+1}+\phi_{i-1,i+1}+\phi_{i-1,i-1}\right) \\
&+\frac{1}{8}D\left[\frac{(\Delta x)^2-3(\Delta y)^2}{(\Delta x)(\Delta y)}\right]\left(\phi_{i,j-1}+\phi_{i,j+1}+\phi_{i,j+1}+\phi_{i,j-1}\right) \\
&+\sigma_a\phi_{i,j}\Delta x\Delta y=S_{i,j}\Delta x\Delta y
\end{aligned}
\tag{3.63}
$$

The off-diagonal elements of the matrix, represented by the second, third and fourth term in Eq.(3.63), are negative when $\Delta x \approx \Delta y$; however, the sign of the second term will become positive if $\frac{\Delta x}{\Delta y} < \sqrt{\frac{1}{3}}$ and the sign of the fourth term will become positive if $\frac{\Delta x}{\Delta y} > \sqrt{3}$. It can be possible to generate unphysically oscillating or even negative solutions when a method's off-diagonal elements are positive. Further analysis and testing are required to determine this. However, for the PWL method, we remark that the second and fourth terms will never both be positive at the same time. For a given source mode, $S(x) = S_w \exp\left(iw_x x + iw_y y\right)$ and $\phi(x) = \phi_w \exp\left(iw_x x + iw_y y\right)$ by definition. These relationships can be substituted into Eq. (3.63) resulting in

$$
\begin{aligned}
&\frac{3}{2}D\left[\frac{(\Delta x)^2+(\Delta y)^2}{(\Delta x)(\Delta y)}\right]\phi_w \\
&+\frac{1}{4}D\left[\frac{(\Delta y)^2-3(\Delta x)^2}{(\Delta x)(\Delta y)}\right]\phi_w\left(e^{iw_y\Delta y}+e^{-iw_y\Delta y}\right) \\
&+\frac{1}{8}D\left[-\frac{(\Delta x)^2+(\Delta y)^2}{(\Delta x)(\Delta y)}\right]\phi_w\left(e^{iw_x\Delta x}+e^{-iw_x\Delta x}\right)\left(e^{iw_y\Delta y}+e^{-iw_y\Delta y}\right)\cdot \\
&+\frac{1}{4}D\left[\frac{(\Delta x)^2-3(\Delta y)^2}{(\Delta x)(\Delta y)}\right]\phi_w\left(e^{iw_x\Delta x}+e^{-iw_x\Delta x}\right) \\
&+\sigma_a\phi_w\Delta x\Delta y=S_w\Delta x\Delta y
\end{aligned}
\tag{3.64}
$$

Eq. (3.64) can be further simplified by noting that $\cos(z)=\frac{1}{2}\left(e^{iz}+e^{-iz}\right)$ and dividing by $\Delta x\Delta y$, which results in

$$
\begin{aligned}
&\frac{3}{2}D\left[\frac{(\Delta x)^2+(\Delta y)^2}{(\Delta x)^2(\Delta y)^2}\right]\phi_w \\
&+\frac{1}{2}D\left[\frac{(\Delta y)^2-3(\Delta x)^2}{(\Delta x)^2(\Delta y)^2}\right]\phi_w\cos\left(w_y\Delta y\right) \\
&+\frac{1}{2}D\left[-\frac{(\Delta x)^2+(\Delta y)^2}{(\Delta x)^2(\Delta y)^2}\right]\phi_w\cos\left(w_x\Delta x\right)\cos\left(w_y\Delta y\right)\cdot \\
&+\frac{1}{2}D\left[\frac{(\Delta x)^2-3(\Delta y)^2}{(\Delta x)^2(\Delta y)^2}\right]\phi_w\cos\left(w_x\Delta x\right) \\
&+\sigma_a\phi_w=S_w
\end{aligned}
\tag{3.65}
$$

Solving for ϕ_w results in

$$\phi_w = \frac{2S_w}{\left\{ D\left[\begin{array}{l} \dfrac{3}{(\Delta y)^2} + \dfrac{3}{(\Delta x)^2} + \left(\dfrac{1}{(\Delta x)^2} - \dfrac{3}{(\Delta y)^2} \right) \cos\left(w_y \Delta y\right) \\ -\left(\dfrac{1}{(\Delta y)^2} + \dfrac{1}{(\Delta x)^2} \right) \cos\left(w_x \Delta x\right) \cos\left(w_y \Delta y\right) \\ +\left(\dfrac{1}{(\Delta y)^2} - \dfrac{3}{(\Delta x)^2} \right) \cos\left(w_x \Delta x\right) \end{array} \right] + 2\sigma_a \right\}}. \qquad (3.66)$$

If $\theta_x = w_x \Delta x$ and $\theta_y = w_y \Delta y$ then Eq. (3.66) becomes

$$\phi_w = \frac{S_w}{D\left[w_x^2 f_{LPWL}\left(\theta_x, \theta_y\right) + w_y^2 f_{LPWL}\left(\theta_y, \theta_x\right) \right] + \sigma_a} \qquad (3.67)$$

where the *f* function for the lumped PWL method is

$$f_{LPWL}\left(\theta_x, \theta_y\right) = \left[2\left(\frac{1-\cos\left(\theta_x\right)}{\theta_x^2} \right) \right] \left[\frac{3+\cos\left(\theta_y\right)}{4} \right]. \qquad (3.68)$$

We now examine the limit as $\theta_x \to 0$ and $\theta_y \to 0$, which is the fine mesh limit. In this case, $\cos\left(z\right) \to 1 - \frac{1}{2} z^2$, and

$$\phi_w = \frac{S_w}{\left\{ D\left[\begin{array}{l} w_y^2 \left[2\left(\dfrac{1-\left(1-\frac{1}{2}\theta_y^2\right)}{\theta_y^2} \right) \right] \left[\dfrac{3+\left(1-\frac{1}{2}\theta_x^2\right)}{4} \right] \\ w_x^2 \left[2\left(\dfrac{1-\left(1-\frac{1}{2}\theta_x^2\right)}{\theta_x^2} \right) \right] \left[\dfrac{3+\left(1-\frac{1}{2}\theta_y^2\right)}{4} \right] \end{array} \right] + \sigma_a \right\}}. \qquad (3.69)$$

As $\theta_x \to 0$ and $\theta_y \to 0$, this simplifies to

$$\phi_w = \frac{S_w}{\left\{D\left[w_y^2 + w_x^2\right] + \sigma_a\right\}} \tag{3.70}$$

Eq. (3.70)is identical to Eq. (3.62), which means that the lumped piecewise linear finite element method limits to the analytic solution in the fine mesh limit, as expected.

We also performed this analysis for an unlumped version of the PWL method because we are interested in comparing how lumping affects the accuracy of the method. The two-dimensional unlumped PWL discretizations on an orthogonal mesh can be written as

$$\begin{aligned}
&\left[\frac{3}{2}D\frac{(\Delta x)^2+(\Delta y)^2}{(\Delta x)(\Delta y)}+\frac{44}{96}\sigma_a\Delta x\Delta y\right]\phi_{i,j}\\
&+\left[\frac{1}{8}D\frac{(\Delta y)^2-3(\Delta x)^2}{(\Delta x)(\Delta y)}+\frac{5}{96}\sigma_a\Delta x\Delta y\right]\left(\phi_{i+1,j}+\phi_{i+1,j}+\phi_{i-1,j}+\phi_{i-1,j}\right)\\
&+\left[-\frac{1}{8}D\frac{(\Delta x)^2+(\Delta y)^2}{(\Delta x)(\Delta y)}+\frac{3}{96}\sigma_a\Delta x\Delta y\right]\left(\phi_{i+1,j-1}+\phi_{i+1,j+1}+\phi_{i-1,j+1}+\phi_{i-1,j-1}\right)\\
&+\left[\frac{1}{8}D\frac{(\Delta x)^2-3(\Delta y)^2}{(\Delta x)(\Delta y)}+\frac{5}{96}\sigma_a\Delta x\Delta y\right]\left(\phi_{i,j-1}+\phi_{i,j+1}+\phi_{i,j+1}+\phi_{i,j-1}\right)\\
&=\begin{bmatrix}\frac{44}{96}S_{i,j}+\frac{10}{96}\left(S_{i+1,j}+S_{i-1,j}\right)+\\ \frac{3}{96}\left(S_{i+1,j-1}+S_{i+1,j+1}+S_{i-1,j+1}+S_{i-1,j-1}\right)+\\ \frac{10}{96}\left(S_{i,j-1}+S_{i,j+1}\right)\end{bmatrix}\Delta x\Delta y
\end{aligned} \tag{3.71}$$

For a given mode, $\phi(x)=\phi_w\exp\left(iw_x x+iw_y y\right)$ and $S(x)=S_w\exp\left(iw_x x+iw_y y\right)$, where S_w and ϕ_w are the coefficients of the mode. After algebra similar to that above we obtain:

$$\phi_w = \frac{S_w}{D\left[w_x^2 f_{UPWL}\left(\theta_x,\theta_y\right)+w_y^2 f_{UPWL}\left(\theta_y,\theta_x\right)\right]+\sigma_a} \tag{3.72}$$

Where the *f* function for the unlumped PWL method is:

$$f_{UPWL}\left(\theta_x,\theta_y\right)=\left[2\frac{1-\cos\left(\theta_x\right)}{\theta_x^2}\right]\left[\frac{3}{1+\left[1+\cos\left(\theta_x\right)\right]\frac{5+3\cos(\theta_y)}{6+2\cos(\theta_y)}}\right] \tag{3.73}$$

Although, it is not shown here, in the fine mesh limit, the unlumped PWL method also approaches the analytic solution.

The third method to be analyzed is Palmer's finite volume method applied to the two dimensional rectangular grid. Palmer's method produces a spatial discretization with a five point coupling.

$$\begin{aligned}&2D\left(\frac{\Delta y}{\Delta x}+\frac{\Delta x}{\Delta y}\right)\phi_{i,j}+\frac{1}{2}D\left(-\frac{\Delta y}{\Delta x}\right)\left(\phi_{i,j-1}+\phi_{i,j+1}+\phi_{i,j+1}+\phi_{i,j-1}\right)\\&+\frac{1}{2}D\left(-\frac{\Delta x}{\Delta y}\right)\left(\phi_{i+1,j}+\phi_{i+1,j}+\phi_{i-1,j}+\phi_{i-1,j}\right)\\&+\sigma_a\phi_{i,j}\Delta x\Delta y=S_{i,j}\Delta x\Delta y\end{aligned} \tag{3.74}$$

Notice that the off-diagonal matrix elements, represented by the second and third term in Eq. (3.74), for Palmer's method will never become positive. As a result, Palmer's method, unlike the PWL FEM, is guaranteed a non-negative solution for problems solved on orthogonal grids.

For a given mode, $\phi\left(x\right)=\phi_w\exp\left(iw_x x+iw_y y\right)$ and $S\left(x\right)=S_w\exp\left(iw_x x+iw_y y\right)$, where S_w and ϕ_w are the coefficients of the mode. After relatively simple algebra we find:

$$\phi_w=\frac{S_w}{D\left[w_x^2 f_{Palmer}\left(\theta_x\right)+w_y^2 f_{Palmer}\left(\theta_y\right)\right]+\sigma_a}, \tag{3.75}$$

where the *f* function for Palmer's method is:

$$f_{Palmer}\left(\theta_x,\theta_y\right)=2\left[\frac{1-\cos\left(\theta_x\right)}{\theta_x^2}\right]\ . \tag{3.76}$$

Palmer's method also limits to the analytic solution in the fine-mesh limit for these problems.

Our fourth method is a bilinear continuous finite element method with mass-matrix lumping:

$$\begin{aligned}
&\frac{4}{3}D\left[\frac{\Delta y}{\Delta x}+\frac{\Delta x}{\Delta y}\right]\phi_{i,j} \\
&+\frac{1}{6}D\left[\frac{\Delta y}{\Delta x}-\frac{2\Delta x}{\Delta y}\right]\left(\phi_{i+1,j}+\phi_{i+1,j}+\phi_{i-1,j}+\phi_{i-1,j}\right) \\
&+\frac{1}{6}D\left[-\frac{\Delta y}{\Delta x}-\frac{\Delta x}{\Delta y}\right]\left(\phi_{i+1,j-1}+\phi_{i+1,j+1}+\phi_{i-1,j+1}+\phi_{i-1,j-1}\right) \\
&+\frac{1}{6}D\left[-\frac{2\Delta y}{\Delta x}+\frac{\Delta x}{\Delta y}\right]\left(\phi_{i,j-1}+\phi_{i,j+1}+\phi_{i,j+1}+\phi_{i,j-1}\right) \\
&+\sigma_a\phi_{i,j}\Delta x\Delta y=S_{i,j}\Delta x\Delta y
\end{aligned} \tag{3.77}$$

The off-diagonal elements of the matrix, represented by the second, third and fourth term in Eq. (3.77), are negative when $\Delta x \approx \Delta y$; however the sign of the second term will become positive if $\frac{\Delta x}{\Delta y}<\sqrt{\frac{1}{2}}$ and the sign of the fourth term will become positive if $\frac{\Delta x}{\Delta y}>\sqrt{2}$. From these results is seems that finite element methods are more susceptible to negative or oscillatory solutions than are finite-volume methods.

For a given mode, $\phi(x)=\phi_w\exp\left(iw_x x+iw_y y\right)$ and $S(x)=S_w\exp\left(iw_x x+iw_y y\right)$. The usual algebra yields:

$$\phi_w=\frac{S_w}{D\left[w_x^2 f_{LBL}\left(\theta_x,\theta_y\right)+w_y^2 f_{LBL}\left(\theta_y,\theta_x\right)\right]+\sigma_a}, \tag{3.78}$$

where the *f* function for the lumped BL method is:

$$f_{LBL}\left(\theta_x,\theta_y\right)=\left[2\left(\frac{1-\cos\left(\theta_x\right)}{\theta_x^2}\right)\right]\left[\left(\frac{2+\cos\left(\theta_y\right)}{3}\right)\right] \tag{3.79}$$

As expected, the bilinear continuous finite element method also limits to the analytic solution in the fine-mesh limit for these problems.

Finally, we performed this analysis for an unlumped version of the BLC method. The two-dimensional unlumped BLC discretization can be written as

$$\begin{aligned}
&\left[\frac{4}{3}D\left(\frac{\Delta y}{\Delta x}+\frac{\Delta x}{\Delta y}\right)+\frac{4}{9}\sigma_a\Delta x\Delta y\right]\phi_{i,j}\\
&+\left[\frac{1}{6}D\left(\frac{\Delta y}{\Delta x}-\frac{2\Delta x}{\Delta y}\right)+\frac{1}{18}\sigma_a\Delta x\Delta y\right]\left(\phi_{i+1,j}+\phi_{i+1,j}+\phi_{i-1,j}+\phi_{i-1,j}\right)\\
&+\left[\frac{1}{6}D\left(-\frac{\Delta y}{\Delta x}-\frac{\Delta x}{\Delta y}\right)+\frac{1}{36}\sigma_a\Delta x\Delta y\right]\left(\phi_{i+1,j-1}+\phi_{i+1,j+1}+\phi_{i-1,j+1}+\phi_{i-1,j-1}\right)\\
&+\left[\frac{1}{6}D\left(-\frac{2\Delta y}{\Delta x}+\frac{\Delta x}{\Delta y}\right)+\frac{1}{18}\sigma_a\Delta x\Delta y\right]\left(\phi_{i,j-1}+\phi_{i,j+1}+\phi_{i,j+1}+\phi_{i,j-1}\right)\\
&=\begin{bmatrix}\frac{4}{9}S_{i,j}+\frac{1}{9}\left(S_{i+1,j}+S_{i-1,j}\right)\\ +\frac{1}{36}\left(S_{i+1,j-1}+S_{i+1,j+1}+S_{i-1,j+1}+S_{i-1,j-1}\right)+\frac{1}{9}\left(S_{i,j-1}+S_{i,j+1}\right)\end{bmatrix}\Delta x\Delta y
\end{aligned} \tag{3.80}$$

For a given mode, $\phi\left(x\right)=\phi_w\exp\left(iw_x x+iw_y y\right)$ and $S\left(x\right)=S_w\exp\left(iw_x x+iw_y y\right)$, where S_w and ϕ_w are the coefficients of the mode. Our algebra eventually produces:

$$\phi_w=\frac{S_w}{D\left[w_x^2 f_{UBL}\left(\theta_x,\theta_y\right)+w_y^2 f_{UBL}\left(\theta_y,\theta_x\right)\right]+\sigma_a}, \tag{3.81}$$

where the *f* function for the unlumped BL method is:

$$f_{UBL}\left(\theta_x,\theta_y\right)=\left[2\left(\frac{1-\cos\left(\theta_x\right)}{\theta_x^2}\right)\right]\left[\left(\frac{3}{2+\cos\left(\theta_x\right)}\right)\right] \tag{3.82}$$

Although it is not shown here, in the fine-mesh limit, the unlumped BLC method also approaches the analytic solution.

These equations are the solution for the analytic, lumped PWL, unlumped PWL, Palmer, lumped BLC and unlumped BLC methods, respectively. When the coefficients (the *f* functions) of w_x^2 and w_y^2 approach one, then the method approaches the analytic solution. The method with the coefficients closest to 1 will be the method with the smallest error for a given mode. A solution for each method can be written in the form of:

$$\phi_w = \frac{S_w}{D\left\{w_x^2 f_{method}\left(\theta_x, \theta_y\right) + w_y^2 f_{method}\left(\theta_y, \theta_x\right)\right\} + \sigma_a} \tag{3.83}$$

Each method has its own value of $f_{method}\left(\theta_x, \theta_y\right)$. The values of f are found in Eqs. (3.68), (3.73), (3.76), (3.79), and (3.82). We further define the error in these *f* functions to be

$$\varepsilon_{method} = \left|1 - f\right| \tag{3.84}$$

Note that only Palmer's method and the unlumped BLC method have coefficients that are dependent on only one θ-variable. Also, recall that $\theta_x = w_x \Delta x$ and $\theta_y = w_y \Delta y$. The result of these definitions is the solution is not only dependent on the mesh size, which is expected, but also on the "mode" of the source and solution itself. If a problem has a non-smooth source, high-wavenumber modes (large *w* values) are required to model the source. For this reason, we can get inaccurate results even for very refined meshes if the source is not smooth. In Chapter V, we will examine a test problem where we have observed this behavior in these methods. Figures 3.2, 3.3, 3.4, 3.5 and 3.6 show plots of $\varepsilon_{method}\left(\theta_x, \theta_y\right)$ for each method. Also, because both Palmer's method and the UBLC method are only dependent on one θ-variable, we have included a plot of both errors on one figure, in Figure 3.7.

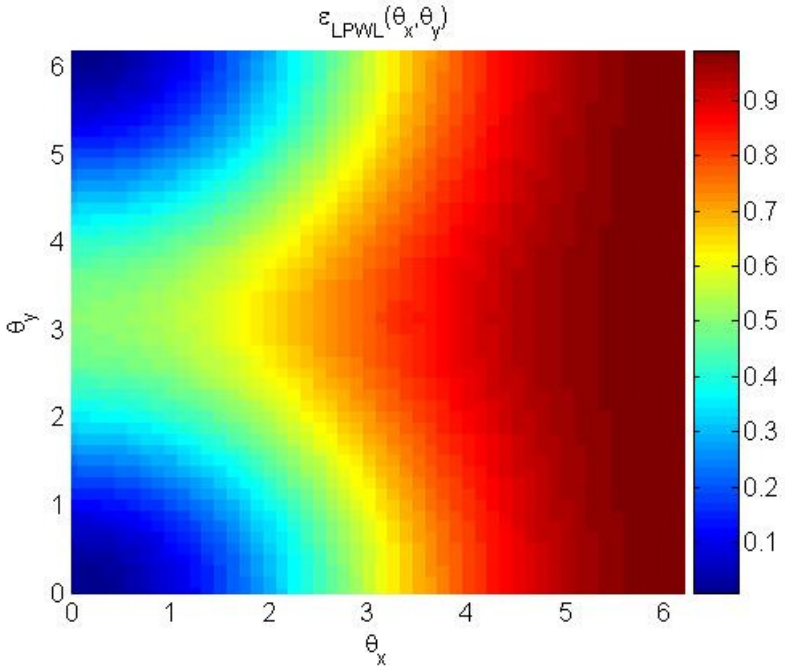


Figure 3.2: Error for lumped PWL

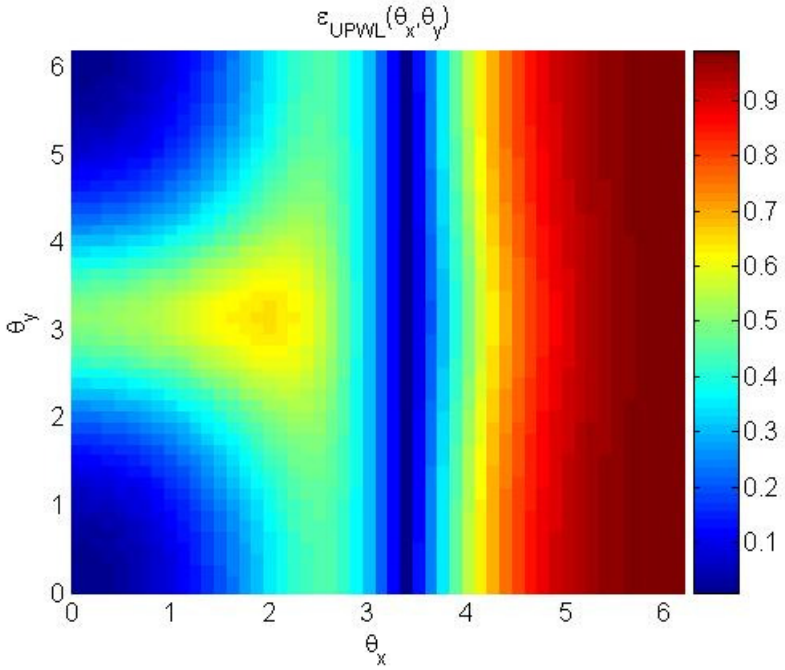


Figure 3.3: Error for unlumped PWL

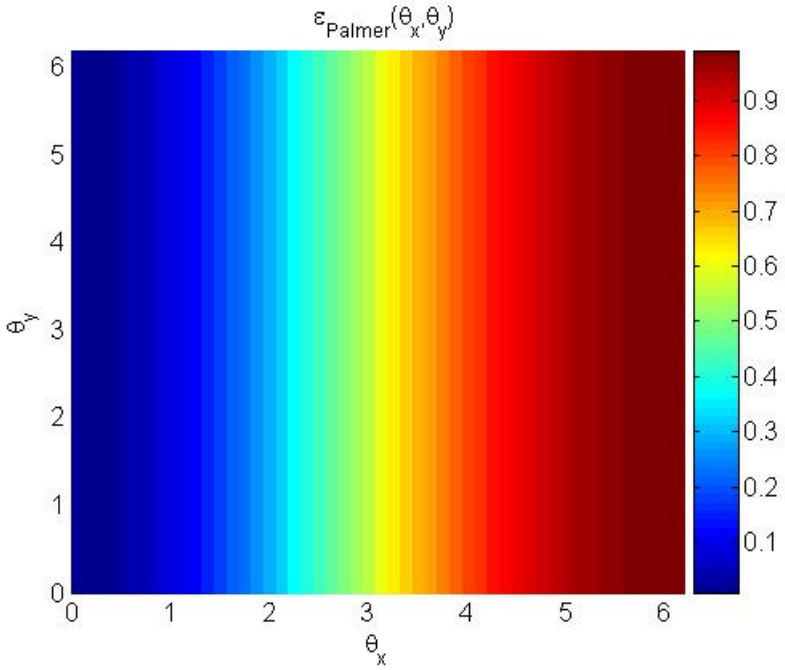


Figure 3.4: Error for Palmer's method

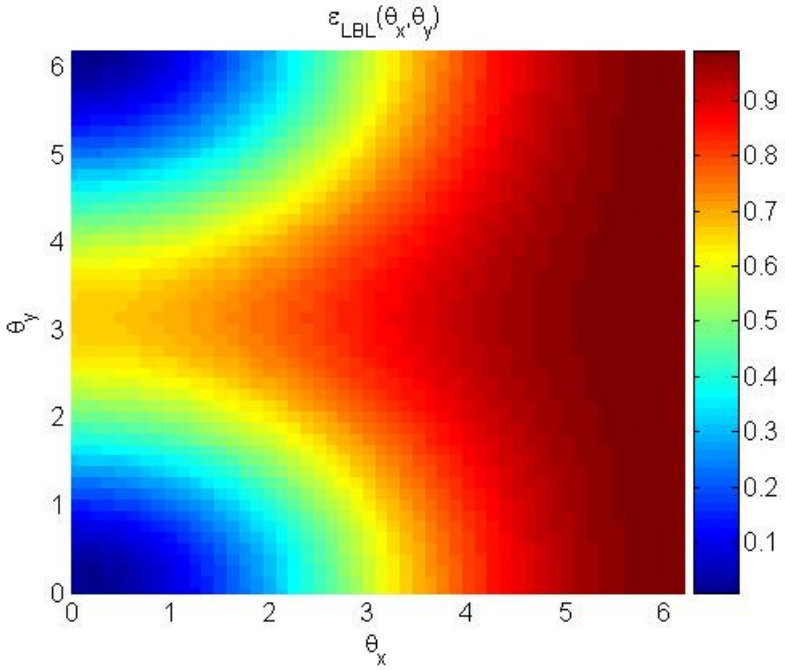


Figure 3.5: Error for lumped bilinear continuous finite element

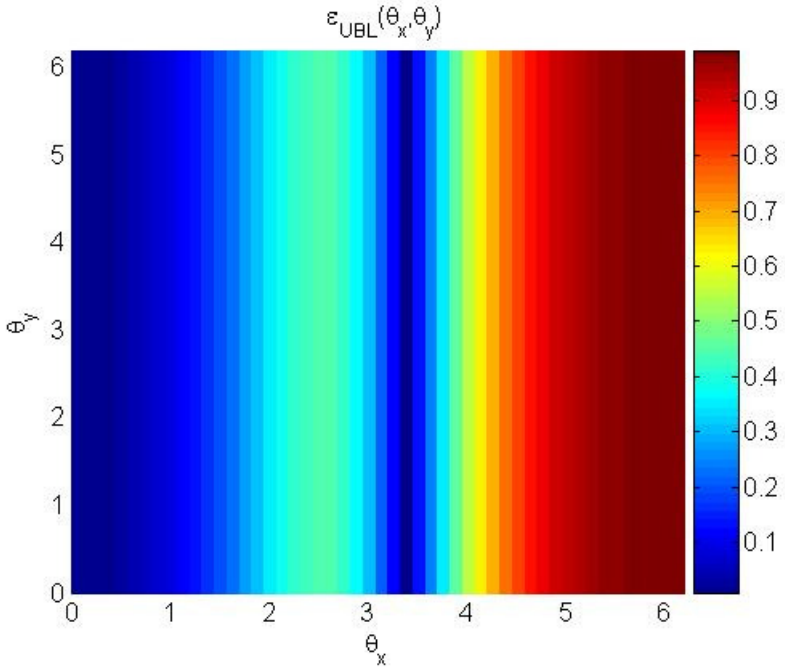


Figure 3.6: Error for unlumped bilinear continuous finite element

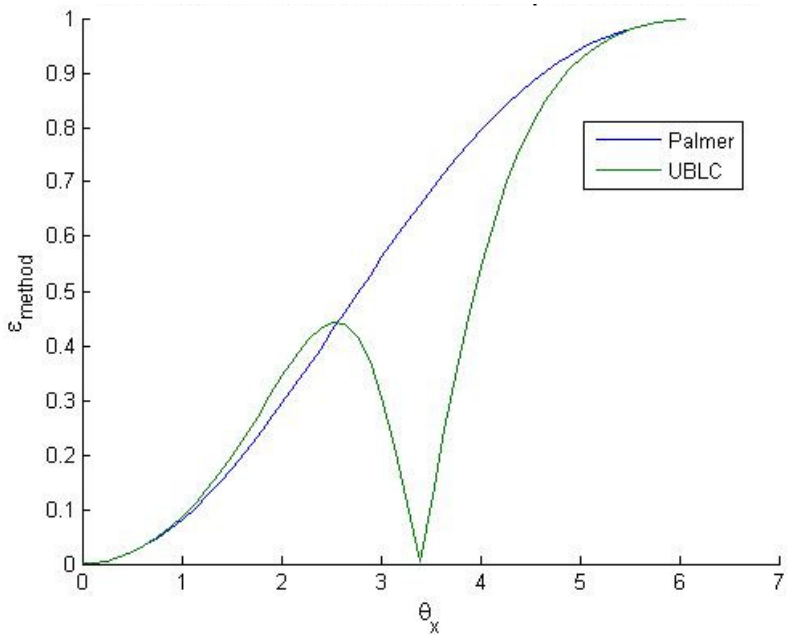


Figure 3.7: Error comparison of Palmer's method and unlumped BLC

From these figures it is easy to see that Palmer's method's coefficients will always be closer to one or equally as close to one as both the lumped PWL and lumped BLC FEM, and the *f* values are monotonically decreasing from a value of one as the problems move further away from the fine mesh limit. As a result, for simple homogeneous problems on rectangular grids we expect Palmer's method to be a little bit more accurate than the lumped FEMs for single-mode problems. This result is interesting because we anticipated that Galerkin finite element methods would be more accurate than Petrov-Galerkin finite element methods for discretizing the diffusion operator. When we "unlump" the FEMs, the *f* values in the unlumped cases are no longer monotonically decreasing. They become larger than one, and then begin to decrease as the mesh size moves away from the fine mesh limit. Furthermore, the unlumped FEM *f* values appear to stay closer to one for a larger range of θs. From this result, we conclude that we gain some accuracy if we unlump the FEMs. Finally, we notice that the behaviors of the PWL and BLC methods are strikingly similar, although the accuracy of the lumped PWL method will always be better than the accuracy of the lumped BLC method for a given mode. This similarity indicates that the derivative discontinuities in the PWL basis functions do not have a large impact on the accuracy of the method.

In this chapter, using the Lax-Milgram lemma we have shown that the PWL discretization applied to the diffusion equation produces a well-posed numerical problem. We have also shown that the PWL method will be second-order accurate using the results from the Lax-Milgram lemma. In the last section of the chapter, we described a simple mode analysis and applied this analysis to the analytic diffusion equation and five separate spatial discretizations of the diffusion equation. From this analysis, we learned that these spatial discretizations all approach the analytic solution in the fine mesh limit, but that the solution accuracy is dependent on the wavenumber of the source as well as the refinement of the mesh. Our mode analysis suggests that for simple problems Palmer's method is likely to be slightly more accurate than the lumped versions of the FEMs. However, we gain accuracy when we unlump the methods. Finally, we learned

that the error in the PWL method appears to be very similar to the error in a BLC method. This modal analysis suggests that the derivative discontinuities in the PWL basis functions do not have a significant impact of the accuracy of the method.

CHAPTER IV
PWL IMPLEMENTATION IN KULL

In Chapter IV, we will present a brief description of the Lawrence Livermore National Laboratory KULL software package and discuss how the PWL method was implemented in the code.

KULL is a massively parallel multi-dimensional multiphysics code project being pursued at Lawrence Livermore National Laboratory. The purpose of this software is to model time-dependent inertial confinement fusion processes and other complex coupled-physics phenomena such as stellar evolution. This simulation involves modeling the motion of the materials in the problem as they change state from solid to plasma due to pressure, temperature, and density changes. KULL uses an Arbitrary Lagrange-Eulerian approach for material motion. The code must also model photon transport or diffusion, where the source of photons in the problem is x-rays being emitted from the hot material. Conductive heat transfer, plasma physics, fusion reactions, and other physical phenomena are also treated. All physical models in this code are discretized on unstructured meshes of arbitrary polyhedral cells [11].

The primary programming language used in the KULL project is C++, which allows the code to be object-oriented. The code structure also allows some pieces to be written in other languages. For example, some of the transport solvers are written in Fortran90. To allow "computational steering" [12] of the complex simulations, an interface between the C++ software and Python scripts has been developed. This allows the code developers and users to easily run complex test problems. Meshes for these problems can be generated using another software package called Draco.

As we have shown previously, the lumped PWL method is similar to Palmer's method. For this reason, given that Palmer's method was already in KULL, it was relatively

simple to add the PWL method. We chose to *lump* the collision and source terms in our PWL method so that their contribution to the coefficient matrix is calculated exactly as in Palmer's method. Both methods calculate the coefficients for the matrix one vertex, or weight function, at a time, resulting in the building of the matrix one row at a time. In the PWL code, the contributions to the coefficients of a given row due to the leakage, or diffusivity term, are built by looping over the sides of the zones that touch the vertex associated with the row. This loop over sides actually begins as a loop over zones attached to the vertex. Inside the loop over zones is a loop over faces in the zone, and inside the loop over faces is the loop over sides. The code is constructed this way to make calculating the value of α and β efficient. For a given side, the contribution to the coefficients due to the α and β terms in the basis and weight functions are collected in the loop over sides. The gradient due to the β term is added when the loop over sides in a face is completed, while the gradient due to the α term is added when the loop over faces in a zone is completed.

The default boundary conditions for the PWL finite element method are the reflecting boundary conditions, which are "natural" boundary conditions in the derivation of the continuous finite element method. Mixed boundary conditions are similar to the reflecting boundary conditions, with only the source vector and one diagonal element being changed in these problems. When Dirichlet boundary conditions are employed, the code loops over every matrix row (or vertex) connected to the boundary vertex and ensures that the matrix remains symmetric by adding Dirichlet boundary information to the source vector.

Two iterative linear solver libraries are available in KULL: PETSc [13] and HYPRE [14]. Because the original method implemented in KULL is Palmer's method, which produces an asymmetric matrix, the interface between KULL and the linear solver libraries uses a full matrix. Thus, in our implementation of PWL the full matrix is built, and the complete benefits of a symmetric method are not realized.

The solver libraries offer many options for iterative methods and preconditioners. The default used for Palmer's discretization is HYPRE's GMRES preconditioned by algebraic multigrid. These options can also be applied to the Galerkin PWL method. In addition, the Galerkin PWL method can use the more efficient and robust Conjugate Gradient (CG) method, with various options for preconditioners. This ability to use CG instead of GMRES is one of the main advantages of the Galerkin PWL method.

CHAPTER V
TEST PROBLEMS

In this chapter we present a series of test problems we used to ascertain properties and performance of the PWL method and Palmer's method. These test problems include a series of problems with linear solutions, a problem with a non-smooth source and high-aspect-ratio cells, a series of problems to determine the order of convergence of both methods, and a problem that models radiation flow through a crooked duct, called the tophat problem, to compare the effectiveness of various linear solvers applied to both methods.

Problems that have linear solutions

To test our implementation of this method as well as the prediction of perfection for linear solutions, we consider a one-dimensional problem with a linear solution. This problem has no source, no absorption, reflecting boundary conditions for the z and y dimensions, and Dirichlet boundary conditions of $E(0,y,z)=0$ and $E(1,y,z)=1$. We can analytically show that this test problem will have a linear solution. We begin with a one-dimensional steady state form of the diffusion equation:

$$-D\frac{d^2E(x)}{d^2x}+\sigma E = S\,, \tag{5.1}$$

with boundary conditions of

$$\begin{aligned} &E(0)=0\\ &E(1)=1\\ &x\in[0,1] \end{aligned} \tag{5.2}$$

Because there is no absorption in this problem the second term in the left hand side of Eq. (5.1) is zero. The right hand side of this equation is also zero because this problem has no source. As a result, Eq. (5.1) simplifies to

$$-D\frac{d^2E(x)}{d^2x}=0. \tag{5.3}$$

The solution to this differential equation is

$$E(x)=C_1x+C_2. \tag{5.4}$$

When the boundary conditions in Eq. (5.2) are applied to this solution, the result is

$$E(x)=x \tag{5.5}$$

Figures 5.1, 5.2, and 5.3 show contour plots of this solution generated by our Galerkin PWL method on a brick mesh, a random mesh, and a "Z-mesh" [15], respectively. From these plots (and many similar ones), we conclude that the PWL method does reproduce the exact linear solution on polyhedral meshes. This property is also attained by Palmer's method.

A z-mesh problem with a linear solution in *x*, *y*, and *z* was also run to show that the method produces the linear solution on a difficult mesh in all dimensions. This test problem had absorption, a linear source, and applied Dirichlet boundary conditions that enforced a linear solution on each face. Solution contours should be straight diagonal lines, and they are. This solution is shown in Figure 5.4.

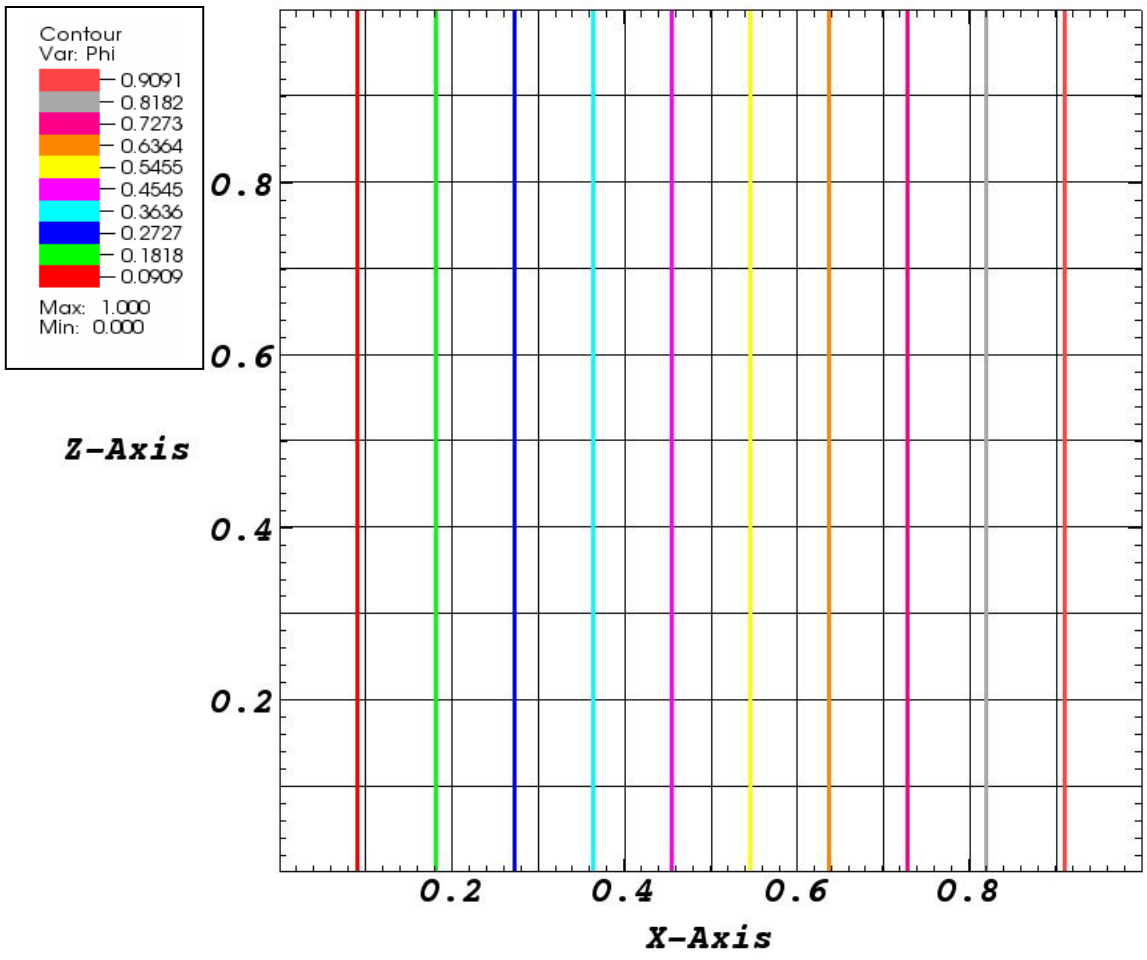


Figure 5.1: Contour plot of a 1D linear solution on an orthogonal mesh at y=0.75

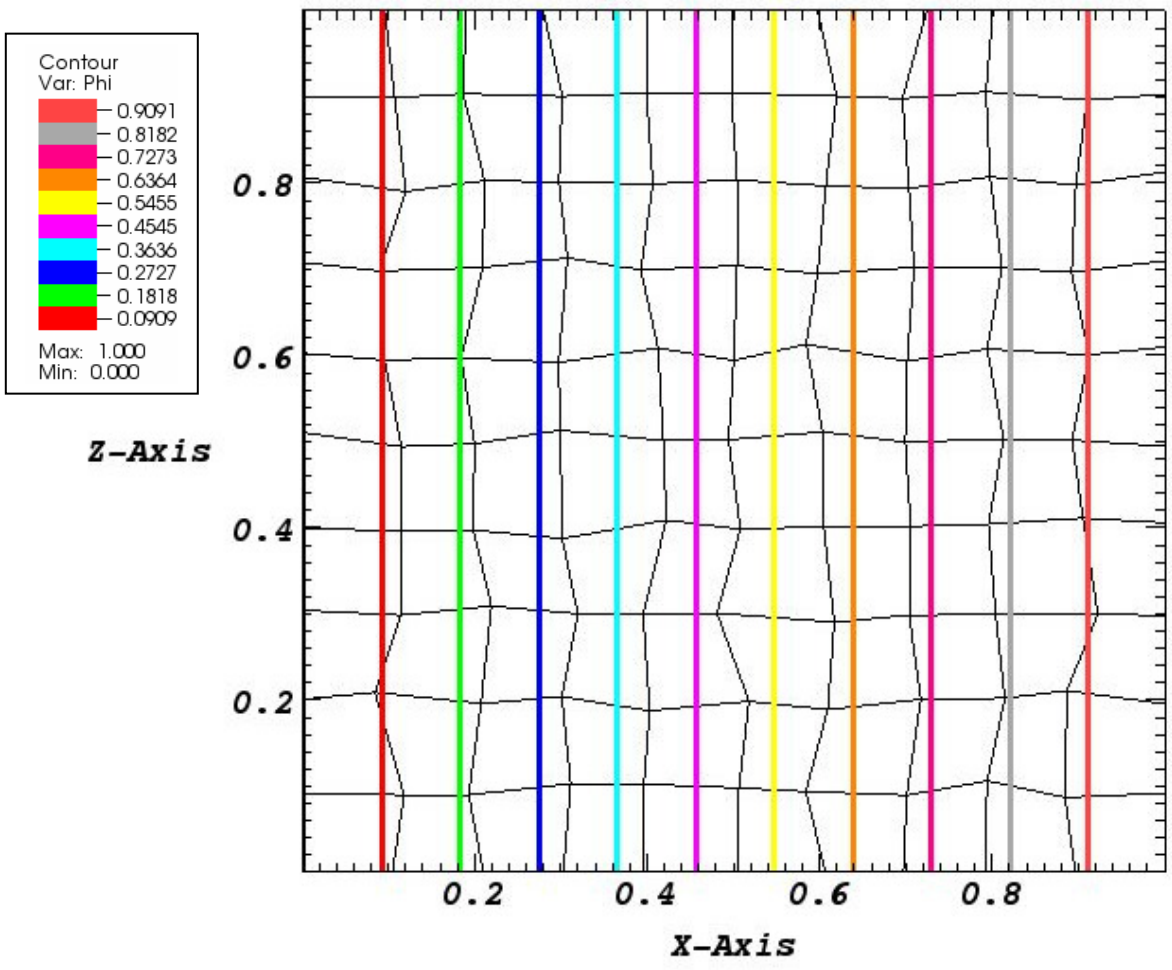


Figure 5.2: Contour plot of a 1D linear solution on a random mesh at y=0.75

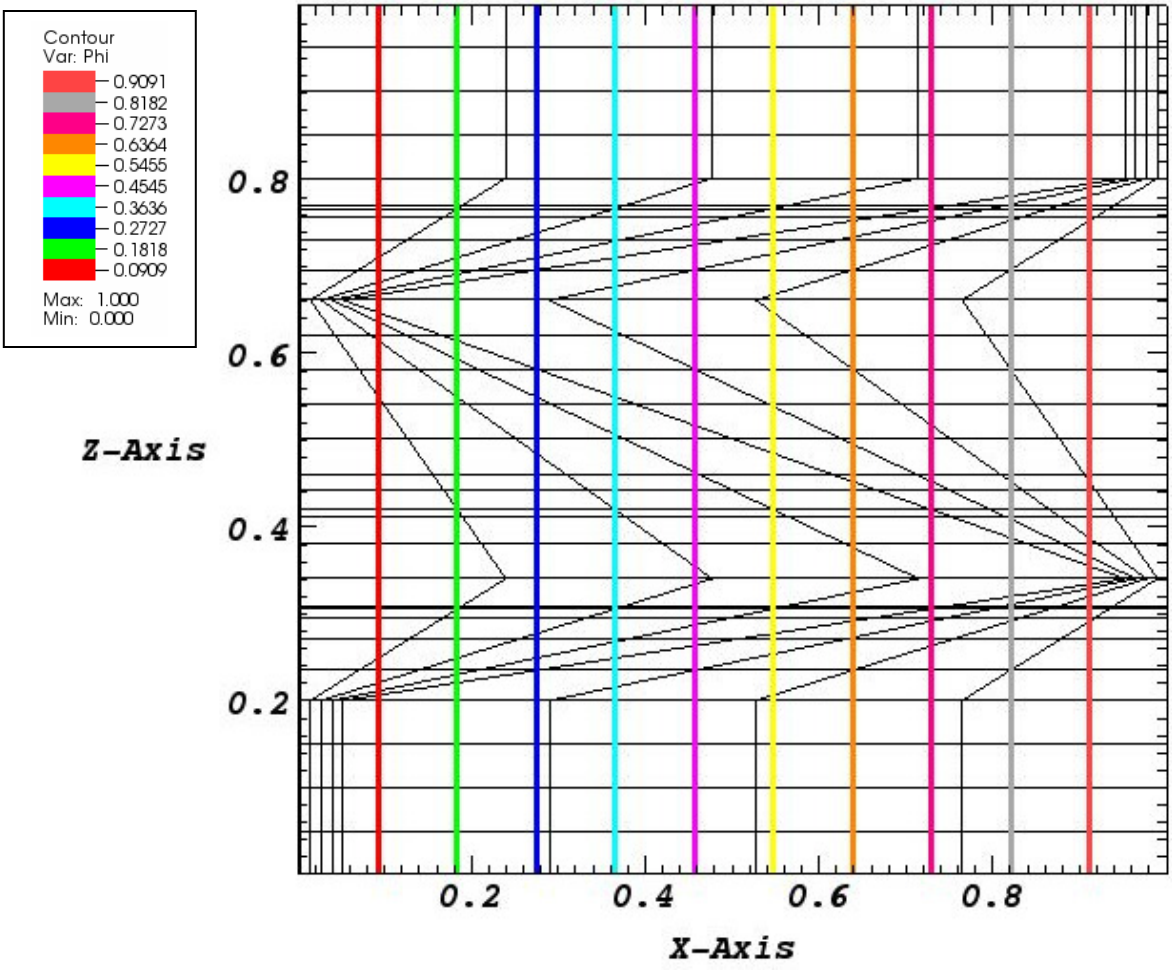


Figure 5.3: Contour plot of a 1D linear solution on a z-mesh at *y*=0.75

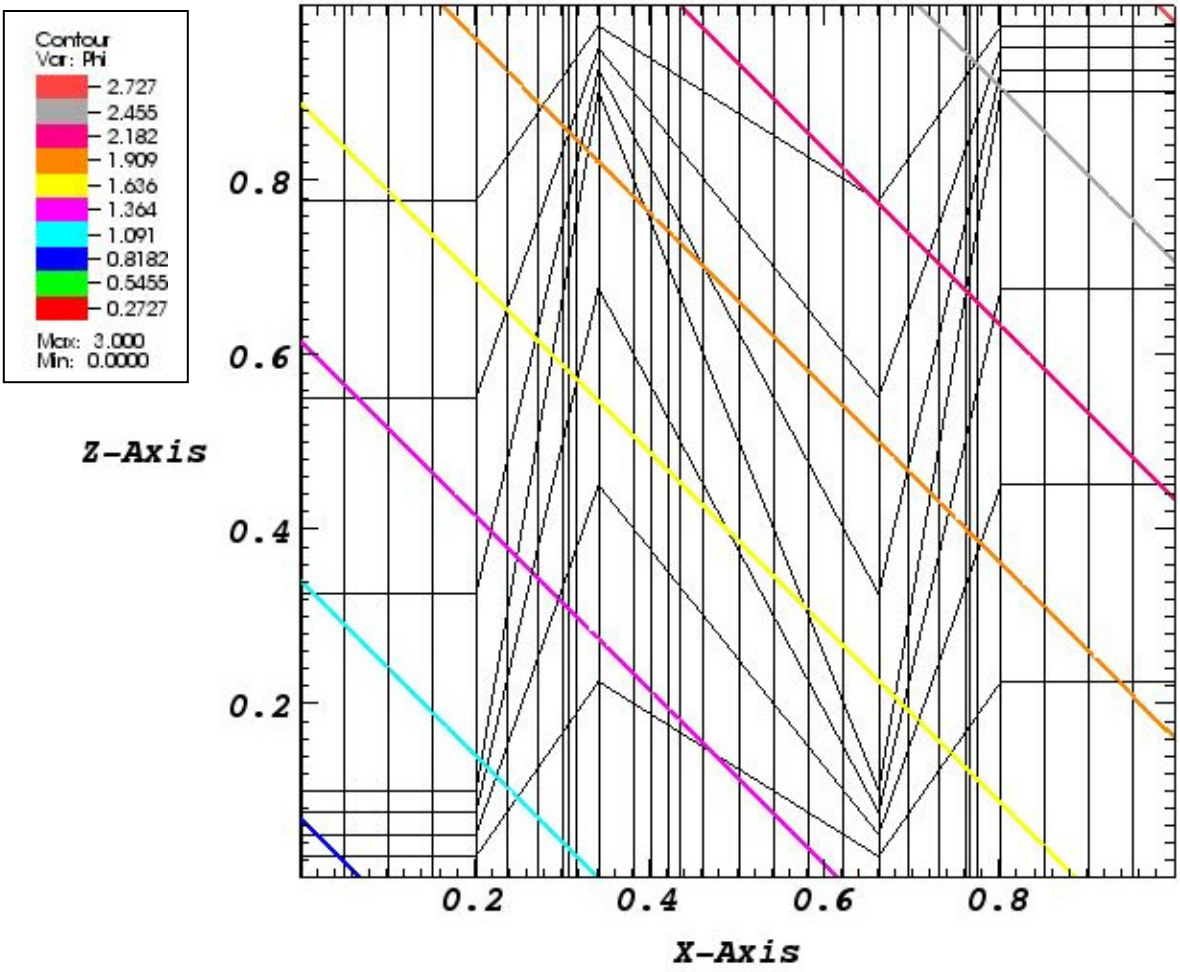


Figure 5.4: Contour plot of 3D linear solution on a *z*-mesh, slice at *y*=0.75

Numerical testing of second-order convergence rate

The next set of problems tests the convergence rate of both the PWL method and Palmer's method on a series of random meshes, and on orthogonal meshes. (Each random mesh was generated by randomly perturbing the vertices in an orthogonal mesh.) One test calculated the convergence rate for a problem with a known one-dimensional quartic solution. This problem has no absorption, a quadratic source, and Dirichlet boundary conditions of $E(0,y,z)=0$ and $E(1,y,z)=1$. The diffusion equation that describes this test problem is

$$-D\frac{d^2E(x)}{d^2x}=x^2, \tag{5.6}$$

with boundary conditions of

$$\begin{aligned} &E(0)=0\\ &E(1)=1\\ &x\in[0,1] \end{aligned} \tag{5.7}$$

The general solution to Eq. (5.6) is

$$E(x)=-\frac{1}{12D}x^4+C_1x+C_2. \tag{5.8}$$

When the boundary conditions are applied to Eq. (5.8), the result is

$$E(x)=-\frac{1}{12D}x^4+\left(1+\frac{1}{12D}\right)x \tag{5.9}$$

The results from this test are found in Figure 5.5.

A second test problem was developed to further test the convergence rate. This problem includes absorption, has no source, and Dirichlet boundary conditions of $E(0,y,z)=1$ and $E(1,y,z)=0$. It has an exponential solution and is described by this diffusion equation:

$$-D\frac{d^2E(x)}{d^2x}+\sigma E(x)=0, \tag{5.10}$$

with boundary conditions of

$$\begin{aligned} &E(0)=1\\ &E(1)=0.\\ &x\in[0,1] \end{aligned} \tag{5.11}$$

The general solution to Eq. (5.10) is

$$E(x)=C_1e^{x/L}+C_2e^{-x/L}, \tag{5.12}$$

where L is the diffusion length, which is defined as

$$L=\sqrt{\frac{D}{\sigma}}. \tag{5.13}$$

When the boundary conditions are applied to Eq. (5.12), the solution to this diffusion problem is

$$E(x) = -\left(\frac{e^{-2/L}}{1-e^{-2/L}}\right)e^{x/L} + \left(\frac{1}{1-e^{-2/L}}\right)e^{-x/L}. \tag{5.14}$$

The results from this test are also found in Figure 5.5.

The error norm is calculated by taking the L_2 norm of the vector of the exact solution minus the calculated solution. The convergence rate of the methods is determined by the slopes of the lines in Figure 5.5. If the slope is four, the method has a second-order convergence rate, which means that the error in the solution gets decreased by a factor of four when the mesh is refined by a factor of two. In Figure 5.5 we plotted a reference line with a slope of exactly four to compare the results of the numerical tests. We also plotted errors from a series of test problems, appearing above the black reference line, to determine the convergence rates of PWL and Palmer's method on a random mesh, and PWL on a brick mesh for the quartic solution. These results show that for this test problem, not much accuracy is lost when the mesh changes from brick to random. A third set of plots can also be found in Figure 5.5 below the reference line. These lines represent the convergence rates of PWL and Palmer's method on a random mesh for an exponential solution, and PWL on a brick mesh for an exponential solution. These results show that for this test problem, the methods lose about a factor of three in accuracy when the mesh is changed from a brick mesh to a random mesh. In general, both methods show second-order accuracy and have the same magnitude of error for both the quartic problem and the exponential problem

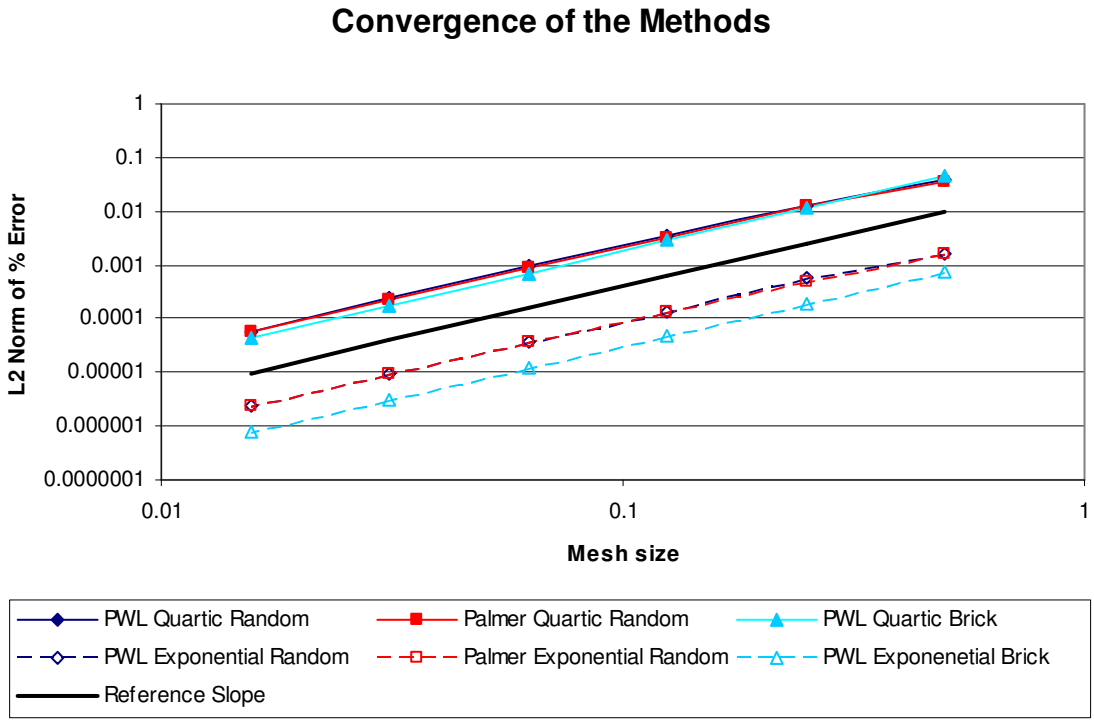


Figure 5.5: Convergence rates of Palmer's method and PWL on various test problems

Problems with high-aspect-ratio cells

In Chapter III, we analyzed the Galerkin PWL method in the limit of high aspect ratios on two-dimensional orthogonal grids, and found that the signs of the off diagonals will change at certain ratios of Δx and Δy. For $(\Delta x/\ \Delta y)^2 < 1/3$, the sign change occurs for the nodes that are above and below the central node. For $(\Delta y/\ \Delta x)^2 > 3$, the sign change occurs for the nodes to the left and right of the central node. We ran a test problem on an orthogonal mesh with 64 cells out of 1024 total cells having 1000 to 1 aspect ratios ($\Delta y/\ \Delta x$). These cells were contiguous, and inserted into the problem at x=0.375 on a cubical domain of unit width. The test problem had Dirichlet boundary conditions of $E(0,y,z)=0$, $E(1,y,z)=0$, $E(x,0,z)=0$ and $E(x,1,z)=0$, and reflecting boundary conditions in

the z direction. The source in this problem is a "point source" (a highly localized source) inserted into the problem in the middle of the cells with high aspect ratios. The plot of the PWL solution is shown for an equally spaced mesh in Figure 5.6 and a mesh with high-aspect-ratio cells in Figure 5.7.

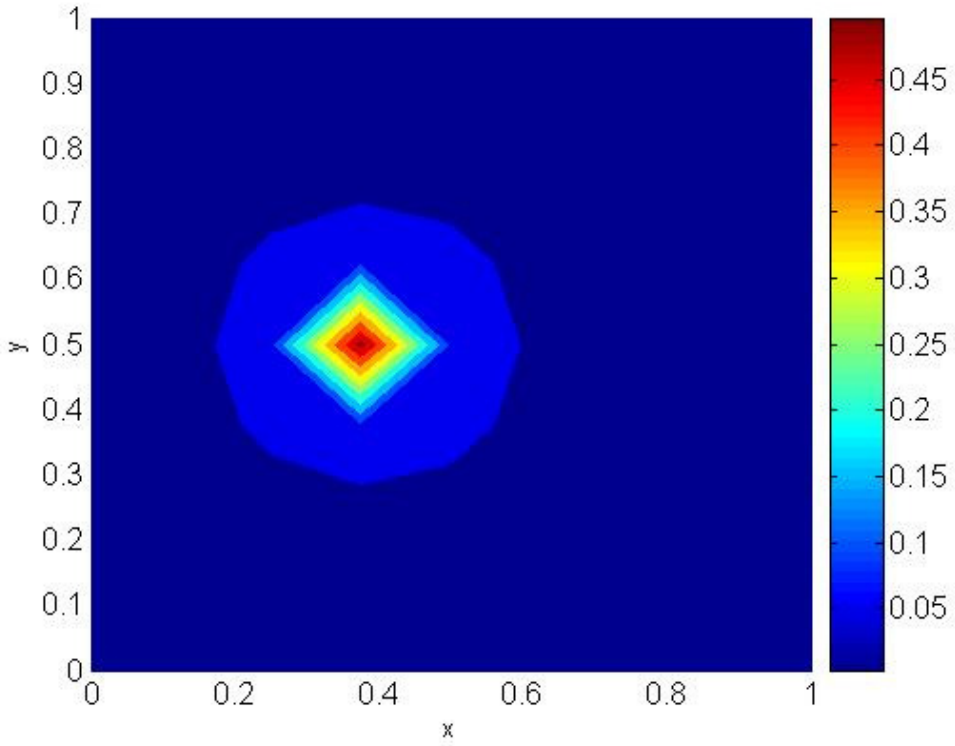


Figure 5.6: Pseudocolor plot of the PWL method at z = 0.5 of the point source problem for an equally spaced mesh with the source at x=0.375 and y=0.5

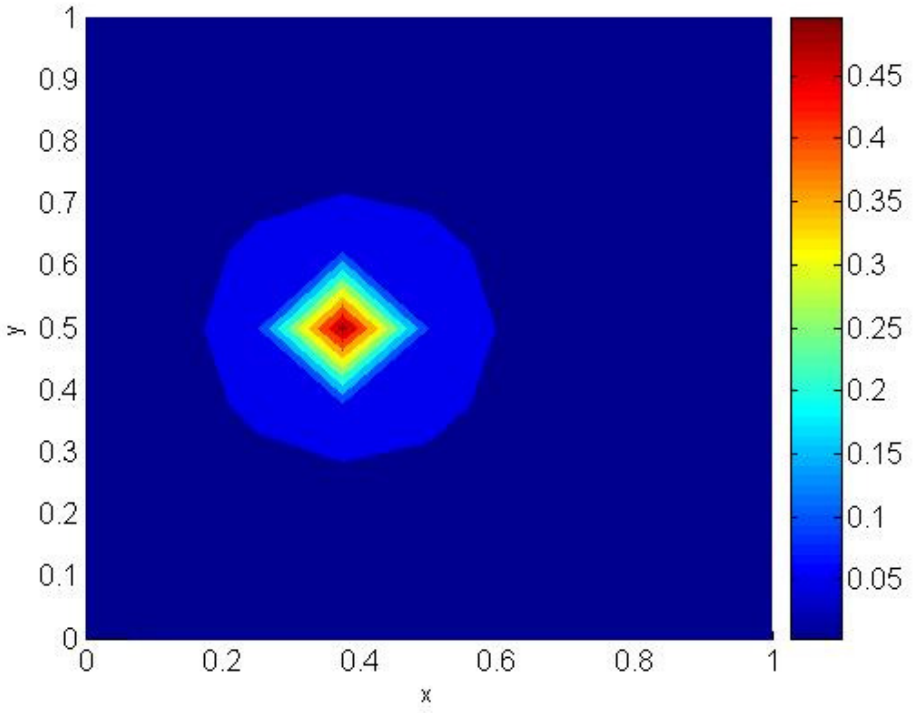


Figure 5.7: Pseudocolor plot of the PWL method at z = 0.5 of the point source problem for a mesh with high-aspect-ratio cells with the source at x=0.375 and y=0.5.

These plots show just a slice of the problem. The slice was taken in the z direction at the value of z where the solution is maximum.

The presence of the high-aspect-ratio cells does not affect the solution significantly. In particular, despite the singular source, there are no unphysical oscillations or negative values in the PWL solution. The problems with uniform and non-uniform grids have the same source strength, which is input into both problems at a single node with the same coordinates. The magnitude of the solution is slightly larger (0.6%) for the problem with the high aspect-ratio cells, but this is simply due to the finer mesh spacing near the source.

We ran this same point source problem on the same meshes using Palmer's method. We do not expect to see negative solutions in the mesh with high-aspect-ratio cells because, as previously noted in Chapter III, Palmer's method is guaranteed a positive solution on orthogonal meshes. The results of this test problem are shown in Figures 5.8, which is an equally spaced mesh and Figure 5.9, which is a mesh that has high-aspect-ratio cells.

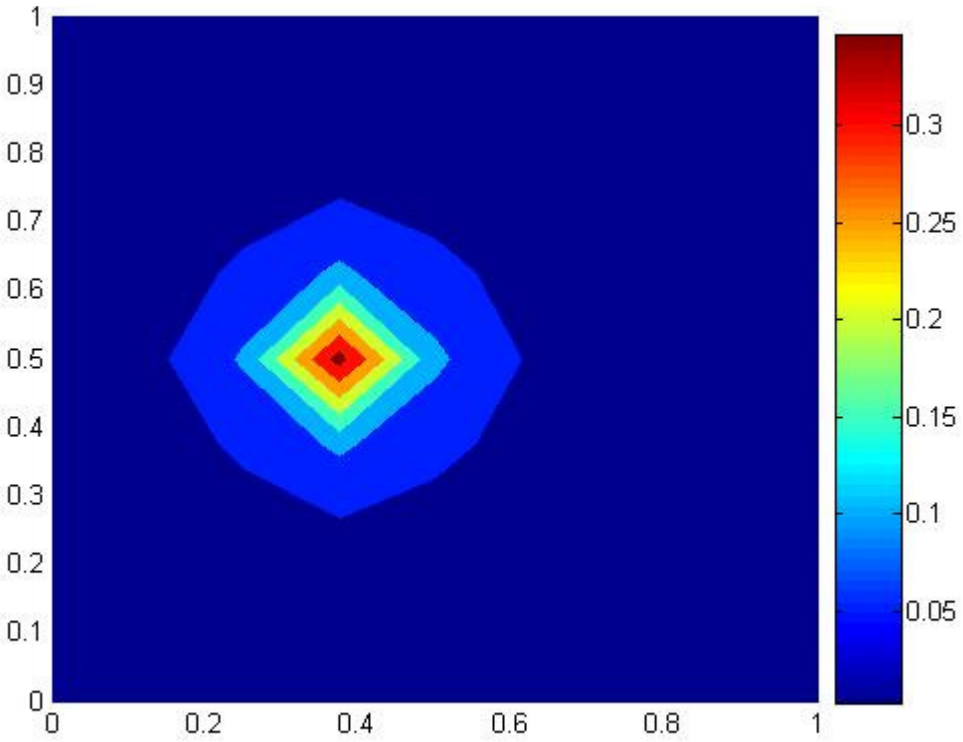


Figure 5.8: Pseudocolor plot of Palmer's method at z = 0.5 of the point source problem for an equally spaced mesh with the source at x=0.375 and y=0.5

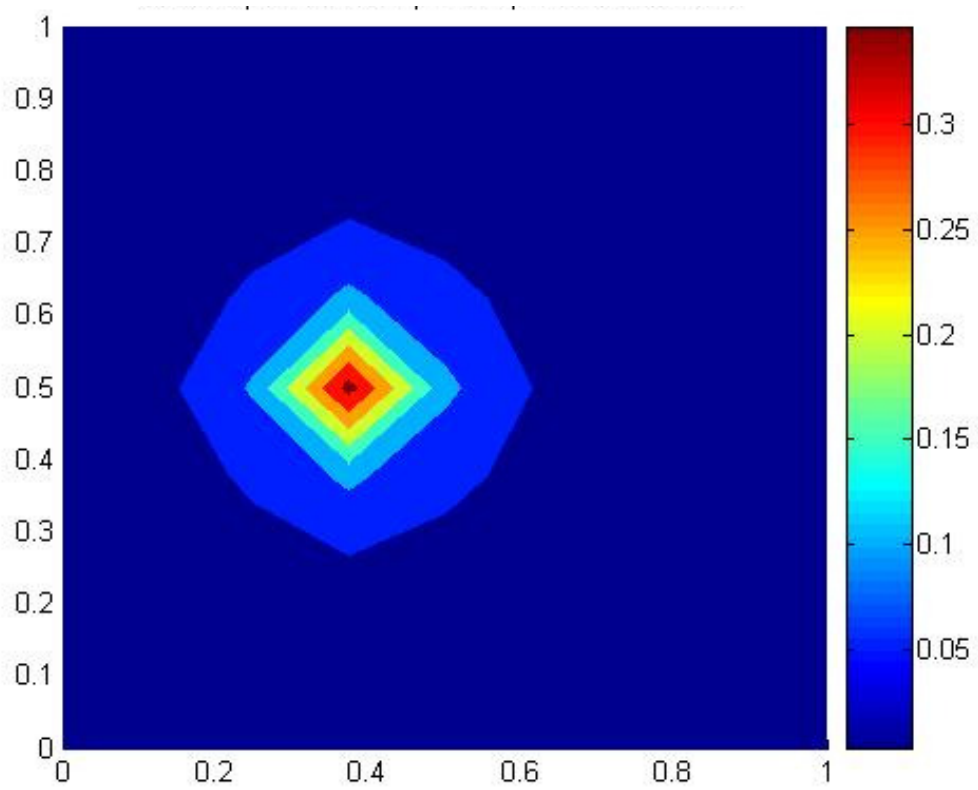


Figure 5.9: Pseudocolor plot of Palmer's method at z = 0.5 of the point source problem for a mesh with high-aspect-ratio cells with the source at x = 0.375 and y = 0.5

For Palmer's method, like PWL, the high-aspect-ratio cell mesh does not affect the solution of the problem, when compared with the solution of Palmer's method on an equally spaced mesh. However, when the solution using Palmer's method is compared with the solution using PWL, we notice that the magnitude of the flux at the location of the point source is significantly different. The PWL value is about 30% greater than the value given by Palmer's method, although away from the source position, the two methods begin to agree. This behavior occurs because, as we have shown in the mode analysis in Chapter III, these methods have difficulty modeling singular sources.

The tophat problem

Because PWL produces an SPD matrix, we can use the Conjugate Gradient (CG) method to solve the matrix system. Palmer's method must be solved using an asymmetric solver such as GMRES. We expect that CG will be more efficient, and we test this conjecture on a time-dependent "tophat" (also known as the crooked duct problem) problem, with the radiation diffusion equation coupled to an energy-balance equation for the material (which the material temperature must satisfy). For this test, the only difference between the two cases is the diffusion solver. Each implementation performs the material temperature coupling in the same way. The tophat problem is a two-material problem, with a high-density region and a low-density tophat region. In Figure 5.10, the red region is the high-density, thick region, and the blue region is the low-density, thin region. While the details are not important for our purposes here, we note that the material opacities are temperature-dependent and the radiation source is a Planckian at the material temperature.

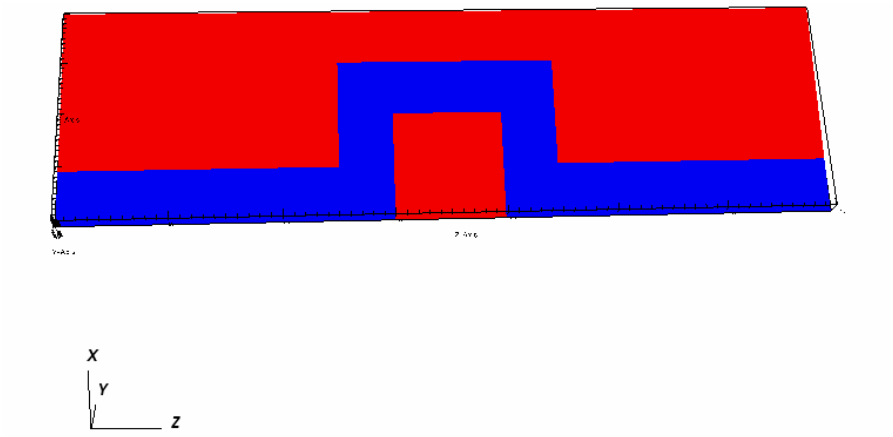
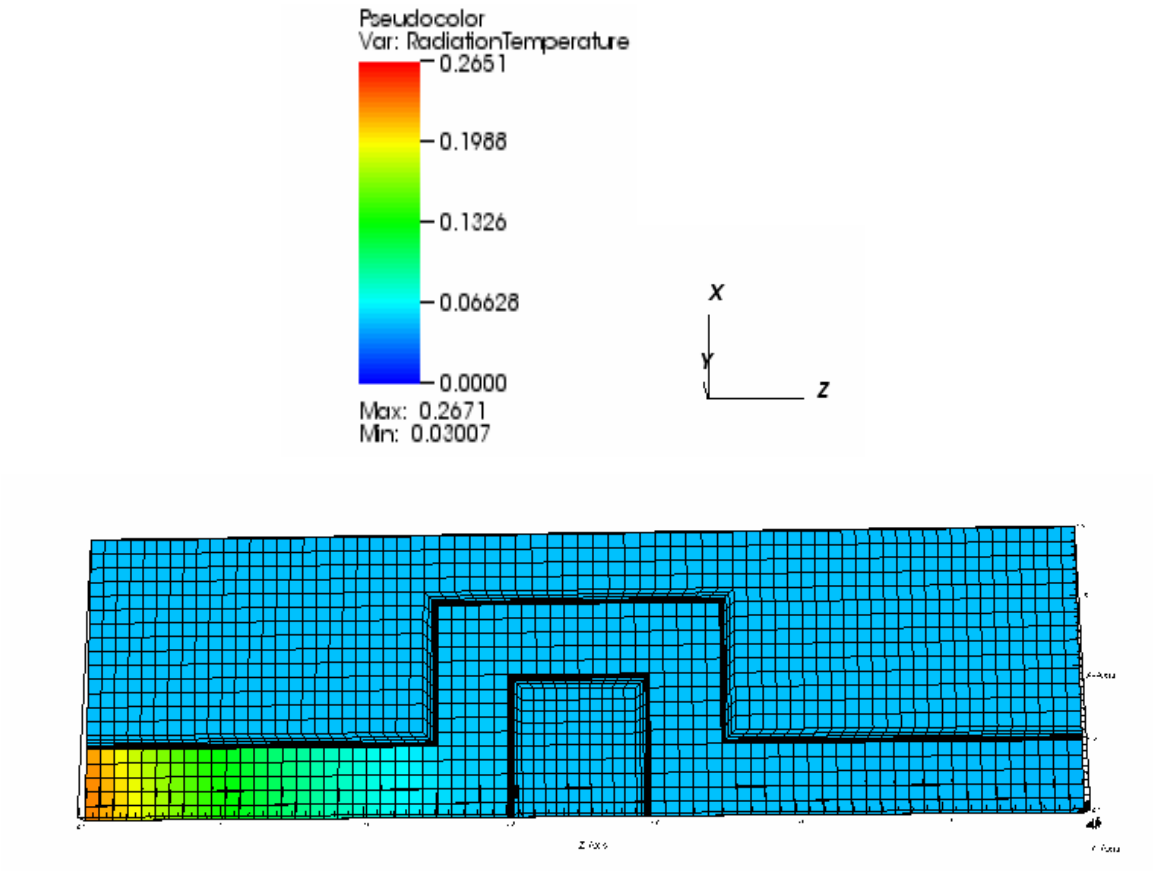


Figure 5.10: The material densities in the tophat problem

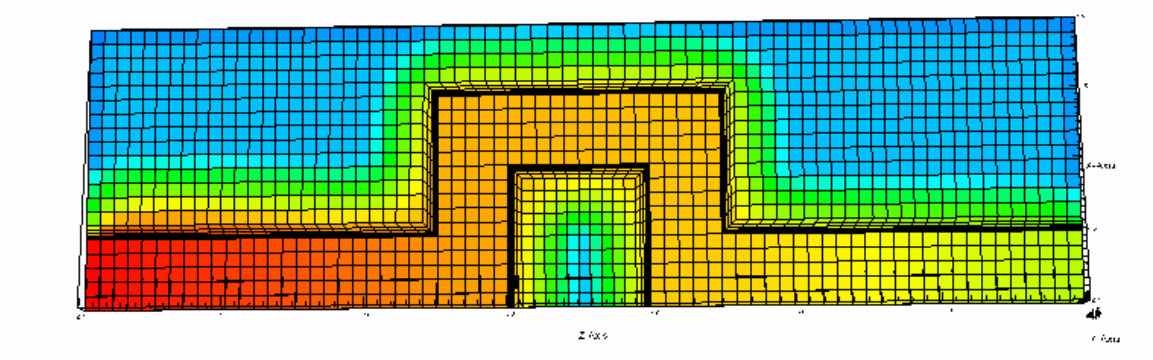
The boundary conditions for this problem are reflecting in the x and y directions and specified intensities in z. An incoming intensity is specified on the left boundary in x along the tophat portion (blue material) of the left surface equivalent to a Planckian

intensity at $kT = 0.3$. Everywhere else on this surface, the value of the boundary condition is vacuum. The initial material temperature is $kT = 0.05$ everywhere in the domain, and the simulation is run for 1000 units of time. A few plots of the simulation results are shown in Figure 5.11. These plots were generated using PWL as the diffusion solver. As expected, the radiation flows through the thin region and also eventually "eats" its way into the thick region. This is interesting in itself, but our purpose here is to illustrate the performance of CG and GMRES solvers.

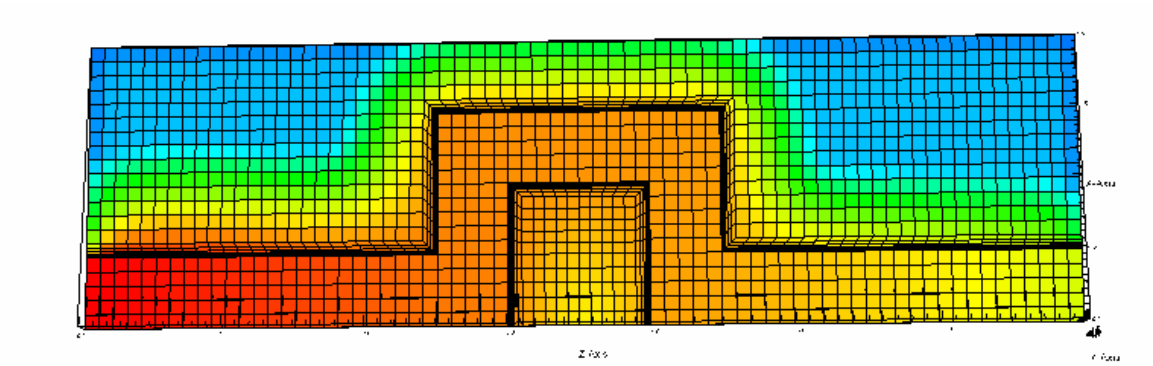


Time = 0.0126066

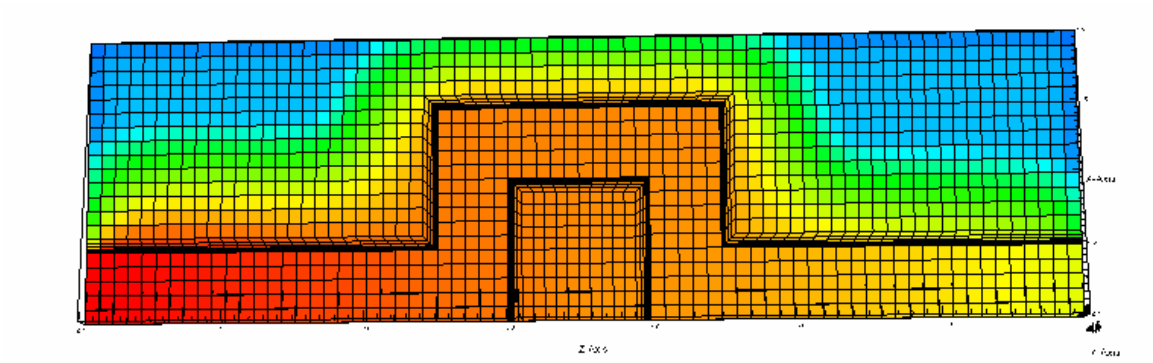
Figure 5.11: Radiation temperature in the tophat problem at different time steps



Time = 348.523



Time = 678.523



Time = 1000

Figure 5.11: Continued

We compared the effectiveness of both iterative methods (GMRES for Palmer and PWL, and CG for PWL) by comparing the number of iterations required to invert the matrix for each reported time step. The preconditioner used for all linear solvers was Algebraic Multigrid.

The results of these calculations, shown in Figure 5.12, show that CG for PWL requires a factor of three fewer iterations than does GMRES for Palmer's method. (Also, in general GMRES applied to PWL requires slightly fewer iterations than GMRES applied to Palmer's method, which indicates that the PWL matrix is slightly better conditioned than that of Palmer's method.) We further note that in a parallel computing environment, CG has even more advantages over GMRES, because it requires far fewer inner products per iteration, and each inner product requires global communication. Our implementation of the PWL method in the KULL software project does not take full advantage of the features of the method and the matrix preconditioner is not optimized for the method. As a result, we have not been able to demonstrate a faster run time with the PWL method as compared to Palmer's method. However, the combination of reduced storage for the matrix, far fewer iterations, reduced storage of solution-length vectors, and fewer inner products per iteration should make the Galerkin PWL method substantially more efficient than Palmer's method when the PWL method is optimally implemented.

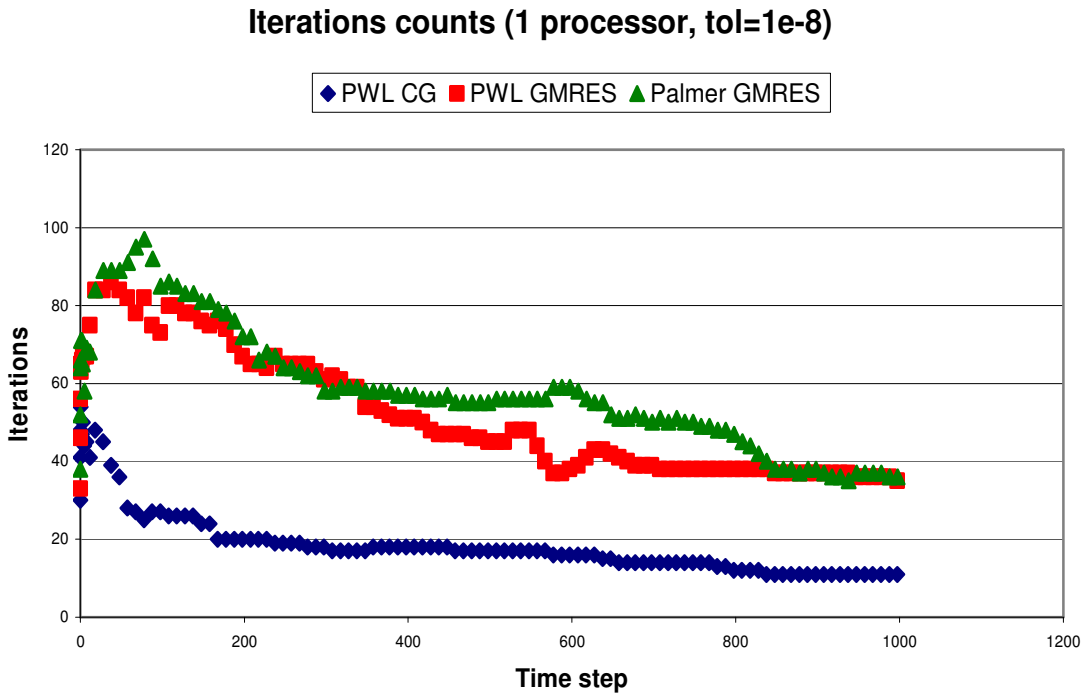


Figure 5.12: Linear solver comparisons for PWL and Palmer's method for one processor

In this chapter, we have presented a series of test problems that show many different properties of the PWL method and Palmer's method. We started by presenting test problems that had linear solutions. Because the PWL method solves these problems exactly, we believe that we coded the method correctly. Also, this test problem shows that the PWL method has similar properties to Palmer's method. Next, we ran a series of test problems to numerically determine the convergence rate of both PWL and Palmer's method. In Chapter III we had predicted that the PWL method would have a second-order convergence rate, which is what the numerical tests showed. The third set of test problems ran problems with non-smooth sources and high-aspect-ratio cells. The purpose of these test problems was to determine if meshes with high-aspect-ratio cells would detrimentally affect the solution. We found that the PWL method and Palmer's method were robust for these test problems. In the last test problem, we developed a

coupled-physics time-dependent problem. Successfully running this test problem demonstrates that the PWL method, like Palmer's method, can handle physically complex problems. For this problem, we also compared the number of iterations for PWL with CG and Palmer's method with GMRES required to solve the coefficient matrices. At every time step, PWL with CG required only a third the number of iterations as Palmer's method with GMRES. This test problem demonstrates the potential computational advantage that the PWL method has due to the symmetry of it coefficient matrix.

CHAPTER VI
CONCLUSION

Summary of the thesis

In this thesis, we have described the application of a Galerkin piecewise linear finite element method to the diffusion equation on arbitrary polyhedral meshes. The method is vertex-centered and is exact for problems with linear solutions. The PWL basis functions are built by dividing an arbitrary polyhedral cell into "side" sub-cell volumes, which are tetrahedra, and adding weighted contributions of the standard linear basis functions on a tetrahedron to create a PWL basis function. The PWL finite element method is guaranteed to produce a symmetric positive definite coefficient matrix; other methods, in particular Palmer's finite volume method, are not guaranteed to be symmetric. We have also shown that Palmer's method is actually a Petrov-Galerkin PWL finite element method.

We then performed some simple mathematical analyses on the PWL method and Palmer's method. We showed that the PWL method is a well-posed discrete problem that has a second-order convergence rate. We also performed a simple mode analysis on the mass-matrix lumped and unlumped PWL methods, Palmer's method, and the lumped and unlumped bilinear continuous finite element methods. This analysis gave us insight into the accuracy of each of these methods, and showed that each of these methods approached the analytic solution in the fine mesh limit.

Our numerical tests showed that the Galerkin PWL finite element method applied to the radiation diffusion equation on arbitrary polyhedral meshes has great potential for computational improvements over previous vertex-centered methods. As we had hoped, the PWL method produces results and behaviors almost identical to those of Palmer's method. We have shown that the PWL exactly reproduces linear solutions for an array

of test problems. We have also shown that PWL and Palmer's method are second-order accurate for a problem with a quartic solution and a problem with an exponential solution on both smooth and non-smooth grids. Palmer's method has a slight theoretical advantage on orthogonal ("brick") grids in that it generates a highly robust seven-point discretization whereas PWL generates full 27-point coupling.

For our test problems with non-smooth "point" sources and very high-aspect-ratio cells on an orthogonal mesh the PWL method did not produce any negative solutions even though some of its off-diagonal matrix values became positive. Finally, we noted because the PWL coefficient matrix is SPD it requires less storage and is potentially much more computationally efficient. We tested this potential using the "tophat" or crooked duct test problem. This problem is a time-dependent radiation-flow problem coupled to an energy-balance equation for the matter. To determine the potential computational efficiency of each method, we counted the number of iterations used to find a matrix solution using its linear solver. Because Palmer's method creates an asymmetric coefficient matrix, GMRES was used as its matrix solver. Conjugate Gradient was used as PWL's matrix solver. The result of the tophat simulation was that the PWL method required a third of the number of iterations at each time step to solve its matrix when compared to Palmer's method.

We conclude that the Galerkin PWL finite element method is a very attractive vertex-centered option for solving diffusion problems on unstructured grids. The PWL method retains the accuracy properties of Palmer's method, including the ability to exactly represent linear solutions, a second-order convergence rate, the ability to solve problems on grids with high-aspect-ratio cells, and the ability to accurately solve physically complex problems. Furthermore, our numerical results and the properties of the PWL matrix strongly imply that given an optimal implementation, PWL will be more computationally efficient than Palmer's method.

Suggestions for future work

A few fundamental improvements can be made in the implementation of the PWL method in KULL. First, it would be beneficial to unlump the PWL method (and also Palmer's method) to see if we gain accuracy, which is predicted by the mode analysis performed in Chapter III. Also, in order to compare run times of Palmer's method and PWL, the preconditioner for the conjugate gradient solver must be optimized.

In the scope of this work, we compared the PWL method and Palmer's method in analytical and numerical tests. We suggest that these vertex-centered methods be further compared to cell-centered methods to determine the accuracy, robustness, and computational expense of each method and each class of methods. It would be beneficial and interesting to be able to determine classes of problems where one method or one set of methods performs significantly better than the other methods.

Another interesting future project would be to develop higher-order piecewise functions that can be applied to arbitrary polyhedral grids. Another would be to apply the piecewise methods to problems that utilize adaptive mesh refinement. We believe that the piecewise linear basis functions are uniquely suited to adaptive mesh refinement problems because the PWL basis functions are well defined for cells with "hanging" nodes.

REFERENCES

1. H.G. Stone, M.L. Adams, A Piecewise Linear Finite Element Basis with Application to Particle Transport, in: Proc. ANS Topical Meeting Nuclear Mathematical and Computational Sciences Meeting, April 6-11, 2003, Gatlinburg, TN, CD-ROM.

2. T.S. Palmer, Discretizing the Diffusion Equation on Unstructured Polygonal Meshes in Two Dimensions, Annals of Nuclear Energy 28 (2000) 1851.

3. T.S. Palmer, A Point-Centered Diffusion Differencing for Unstructured Meshes in 3-D, in: Proc. Int. Conf. Mathematics and Computations, Reactor Physics and Environmental Analyses, April 30-May 4, 1995, Portland, OR, 2, p. 897.

4. E.L. Wachspress, A Rational Finite Element Basis, Academic Press, New York, 1975.

5. G. Davidson, T.S. Palmer, Finite Element Transport Using Wachspress Rational Basis Functions on Quadrilaterals in Diffusive Regions, in: Proc. ANS Topical Meeting Mathematics and Computation, Supercomputing Reactor Physics and Nuclear and Biological Applications, September 12-15, 2005, Avignon, France, CD-ROM.

6. J.E. Morel, A 3-D Cell-Centered Diffusion Discretization for Arbitrary Polyhedral Meshes, Los Alamos National Laboratory Research Note, CCS-4:02-40(U), 2002.

7. Y. Kuznetsov, K. Lipnikov, M. Shashkov, Mimetic Finite Difference Method on Polygonal Meshes, Los Alamos National Laboratory Report, LA-UR-03-7608, 2003.

8. E.W. Larsen, G.C. Pomraning, V.C. Badham, Asymptotic Analysis of Radiative Transfer Problems, J. Quant. Spectrosc. Radiat. Transfer 29 (1983) 285.

9. M.L. Adams, P.F. Nowak, Asymptotic Analysis of a Computational Method for Time- and Frequency- Dependent Radiative Transfer, J. Comput. Phys. 146 (1998) 366.

10. A. Ern , J.L. Guermond, Theory and Practice of Finite Elements Springer-Verlag, New York, 2004.

11. J.A. Rathkopf, D.S. Miller, J.M. Owen, L.M. Stuart, M.R. Zika, P.G. Eltgroth, N.K. Madsen, K.P. McCandless, P.F. Nowak, M.K. Nemanic, N.A. Gentile, N.D. Keen, T.S. Palmer, KULL: LLNL's ASCI Inertial Confinement Fusion Simulation Code, Physor 2000 ANS Topical Meeting, May 7-11, 2000, Pittsburgh, PA.

12. T.-Y.B. Yang, D.M. Beazley, P.F. Dubois, G. Furnish, Steering Object-Oriented Computations in Physics, Python Conference, November 3-5, 1996, Washington, DC.

13. S. Balay, W.D. Gropp, L.C. McInnes, B.F. Smith, Efficient Management of Parallelism, in: Object-Oriented Numerical Software Libraries, Birkhauser Press, Boston, MA, 1997.

14. R.D. Falgout and U.M. Yang, HYPRE: a Library of High Performance Preconditioners, Lawrence Livermore National Laboratory Technical Report, UCRL-JC-146175 (2002).

15. D.S. Kershaw, Differencing of the Diffusion Equation in Lagrangian Hydrodynamic Codes, J. Comput. Phys. 39 (1981) 375.

VITA

Name:	Teresa S. Bailey
Address:	Department of Nuclear Engineering, Texas A&M University, 3133 TAMU, College Station, TX 77843-3133
Email Address:	baileyte@tamu.edu
Education:	M.S., Nuclear Engineering, Texas A&M University, 2006 B.S., Nuclear Engineering, Oregon State University, 2002